UWS UNIVERSITY OF THE WEST of SCOTLAND

UWS Assessment Student Declaration

This Declaration must be completed and submitted along with your assessment. Please ensure sections 1 and 2 are completed before submission.

- For written assessments, insert the Declaration as the first page of your document.
- For assessments not in the written format (e.g. video, audio, presentation, or practical work), submit the Declaration as a separate file.
- For group assessments, unless otherwise directed by your lecturer, the group may submit a single shared declaration.

1. Declaration that this is your OWN work

All UWS students are expected to uphold the values of academic integrity: UWS Student Academic Integrity Procedure.

Academic Integrity means a commitment to, and upholding of, the values of honesty, trust, fairness, respect, responsibility and courage in learning, teaching, research and engagement with the University community.

You are responsible for ensuring that the work you submit is your own, and that any use of Generative AI follows the guidance provided for the assessment.

I confirm this assessment is my own work and complies with the guidance provided.	**YES**

2. Declaration of appropriate use of Generative AI

You must check the assessment guidance to understand what uses, if any, of Generative AI are permitted for this assessment.

I used Generative AI for this assessment.	**YES**

(This part is copied from the extensive explanation an the beginning of my thesis):

Statement on the Use of Generative AI

This thesis used Generative AI (GAI) tools—specifically ChatGPT (https://chat.openai.com/), Claude (https://claude.ai/), and DeepSeek (https://www.deepseek.com/)—between 2023 and 2025 as assistive systems for writing refinement and software scaffolding. The author is solely responsible for the scientific ideas, the experimental design, the data processing, the analysis and interpretation of results, and all conclusions. Model outputs were treated as drafts subject to verification; no confidential or proprietary data were uploaded to these systems.

All text and code generated with the assistance of GAI tools was critically reviewed, revised where necessary, and fully validated by the author. The author takes complete responsibility for the correctness, originality, and academic integrity of the final manuscript and experimental implementations. As GAI outputs may contain inaccuracies or unverifiable statements, they were used only as provisional drafts subject to systematic checking.

Writing. The role of GAI in the writing process was strictly limited to style improvements, shortening overly long passages, suggesting clearer structures, and maintaining consistent English grammar and technical terminology. No scientific thoughts, hypotheses, experimental setups, or references were generated by GAI. Literature research was carried out through web search and deep research functions of the GAIs; all references were read by the author and the author did not rely on the GAI's outputs. Text passages were always provided beforehand by the author, at times in the form of bullet points or rough drafts, and GAI suggestions served only as language refinements.

Programming support. The main use of GAI concerned programming support, primarily for debugging, generating helper or auxiliary functions, and suggesting more elegant implementations of existing code blocks using established libraries or idioms. Drafts produced by GAI were always tested against the author's own implementations to ensure correctness and consistency with the intended design. Tasks included writing small utilities (e.g., data loaders, preprocessing helpers, or reusable plotting functions), improving readability by modularizing repetitive code, and pointing out potential optimizations with standard PyTorch or NumPy functionalities.

Visualization. GAI was also consulted for visualization code, such as preparing figures in matplotlib, creating canvases for embedding projections (e.g., PCA plots), or formatting result tables for clarity. These outputs were treated as scaffolding, with final plots and statistical analyses based entirely on the author's experimental data and manual interpretation. At no point did GAI design new models, invent training paradigms, define loss functions, or set hyperparameters. The conception of architectures (e.g., Stock2Vec, QMSE, recurrent and transformer-based variants, and pretraining objectives), the design of experimental protocols, and all methodological choices remain original contributions of the author.

In short, GAI was employed to accelerate routine programming tasks, to assist in debugging and visualization, and to polish the presentation of results, while the intellectual contributions—including the research framing, model design, training strategies, and empirical evaluation—are entirely the author's own.

Extenuating Circumstances

The University recognises that, from time to time, you may encounter circumstances that affect your ability to complete or submit an assessment. If this happens you can submit an Extenuating Circumstances Submission (ECS). In submitting each piece of coursework or completing an examination or class test, you are confirming that you are 'fit to sit' the assessment and wish that any mark achieved for that assessment should stand. You can submit an ECS up to 48 hours after the assessment deadline, including where you have submitted the assessment but believe your academic performance has been affected by extenuating circumstances. The School Assessment Board will know that an ECS has been submitted when recording your module marks. See the UWS Extenuating Circumstances Submission Procedure for further guidance.

Generalizing Natural Language Processing Strategies for Multivariate Time Series Processing on the Example of Quantitative Stock Data

by

Frederic Voigt

Thesis submitted in partial fulfilment of the requirements
of the University of the West of Scotland
for the award of Doctor of Philosophy

18th October 2025

Declaration

The research presented in this thesis was carried out by the undersigned. No part of the research has been submitted in support of an application for another degree or qualification at this or another university.

Signed:

Date: 15.10.2025

It is well enough that people of the nation do not understand our banking and monetary system, for if they did, I believe there would be a revolution before tomorrow morning.

Henry Ford

Abstract

In recent years, Natural Language Processing (NLP) has evolved into one of the most dynamic and well-resourced fields in artificial intelligence, yielding powerful modeling strategies such as word embeddings, hierarchical representations, and large-scale pre-training. While these innovations have transformed language understanding, their application to other domains remains limited. This thesis is motivated by the ambition to transfer NLP methodologies to a new, structurally analogous domain: multivariate time series forecasting in financial markets.

A compelling testbed for this endeavor is offered by financial markets. Many of the structural properties observed in language—such as temporal order, local and global dependencies, and hierarchical patterns across multiple time resolutions—are exhibited by stock price data. By mapping tokens to price observations, sentences to time windows, and contextual embeddings to multivariate stock representations, key NLP techniques—including Word2Vec, Doc2Vec, Masked Language Modeling, Next Sentence Prediction, hierarchical encoders, and (recurrent) transformers—are systematically adapted to the domain of quantitative finance.

It is hypothesized that these adapted techniques are well-suited to addressing persistent challenges in financial forecasting, most notably distribution shifts, non-stationarity, and data scarcity. By leveraging pre-training on large-scale historical price data and by enabling the learning of context-sensitive representations, improved generalization across changing market regimes and enhanced robustness—even in settings with limited or noisy training signals—can be achieved by the models.

The adapted NLP methods consistently outperform classical baselines in standard forecasting tasks. Pre-trained foundation models tailored to financial data show strong potential—particularly in pre-training performance, cross-market adaptability, and faster fine-tuning. While they are not the most accurate forecasters per se, their strengths as general-purpose representations are better leveraged in tasks like market summarization, cross-market transfer learning, risk modeling, portfolio optimization, and synthetic simulations. Finally, an approach for a generalized transfer of NLP strategies for (all) other multivariate time series is proposed.

Acknowledgements

Thanks to my family, my parents, my sister. Thanks all my supervisors from Hamburg; to Prof. Dr. Kai von Luck and to Prof. Dr. Peer Stelldinger and from Scotland; to Prof. Dr. Qi Wang, to Prof. Dr. Jose Alcaraz Calero and to Prof. Dr. Keshav Dahal. Further I liked to thank the staff at the Promotionszentrum in Hamburg; Inga Hesse, Prof. Dr. Michael Gille and Dr. Andreas Sonntag. And a big thank you to all the members of the ML-AG.

List of Publications

Frederic Voigt. Adapting Natural Language Processing Strategies for Stock Price Prediction. In DC@KI2023: Proceedings of Doctoral Consortium at KI 2023, pages 20–29. Gesellschaft für Informatik e.V., 2023.

Frederic Voigt, Kai von Luck, and Peer Stelldinger. Assessment of the Applicability of Large Language Models for Quantitative Stock Price Prediction. In Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '24), pages 293–302. Association for Computing Machinery, 2024.

Frederic Voigt, José M. Alcaraz Calero, Keshav P. Dahal, Qi Wang, Kai von Luck, and Peer Stelldinger. Towards Machine Learning Based Text Categorization in the Financial Domain. In Proceedings of the 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS), 2024.

Frederic Voigt, José M. Alcaraz Calero, Keshav P. Dahal, Qi Wang, Kai von Luck, and Peer Stelldinger. Adapting Speech Models for Stock Price Prediction. In Proceedings of the 2024 IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), 2024.

Frederic Voigt, José M. Alcaraz Calero, Keshav P. Dahal, Qi Wang, Kai von Luck, and Peer Stelldinger. Quantitative Market Situation Embeddings: Utilizing Doc2Vec Strategies for Stock Data. In Proceedings of the IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr 2024), 2024.

Contents

List of Figures

List of Tables

Acc	Accuracy
AE	Auto Encoders
ADR	American Depositary Receipt
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
ASM	Adapted Speech Model
AV	Alpha Ventage
BERT	Bidirectional Encoder Representations from Transformers
BCE	Binary Cross Entropy
CBOS	Continuous Back of Stocks
CBOW	Continuous Back of Words
CLM	Conditional Language Modeling
CR	Cumulative Return
CRSP	Center for Research in Security Prices
CV	Computer Vision
CWRNN	Clockwork Recurrent Neural Network
DL	Deep Learning
DWT	Discrete Wavelet Transform
EBITA	Earnings Before Interest, Taxes and Amortization
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
EPS	Earnings Per Share
ESG	Environmental, Social and Governance
ETF	Exchange Traded Fund
GARCH	Generalized Auto Regressive Conditional Heteroskedasticity
GNN	Graph Neural Network
HMM	Hidden Markov Models
IPO	Initial Public Offering
IRR	Internal Rate of Return
LLM	Large Language Model

LOB	Limit Order Book
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCC	Matthews Correlations Coefficient
MDD	Maximum Drawdown
ML	Machine Learning
MLM	Masked Language Modeling
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NLP	Natural Language Processing
NSP	Next Sequence Prediction
OECD	Organization for Economic Co-operation and Development
OHCL(V)	Open, High, Close, Low (Volume)
OOD	Out of Distribution
PCA	Principal Component Analysis
RL	Reinforcement Learning
RLR	Relative Logarithmic Return
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
RWT	Random Walk Theory
S2V	Stock2Vec
SDM	Stock Distribution Modeling $\in$ SF
SF	Stock Forecasting $\supseteq \{\text{SMP}, \text{SPP}, \text{SDM}\}$
SG	Skip Grad
SGD	Stochastic Gradient Descent
SMA	Simple Moving Average
(s)MAPE	(Symmetric) Mean Absolute Percentage Error
SMC	Stock Movement Classification $\notin$ SF
SMP	Stock Movement Prediction $\in$ SF

SOTA	State of the Art
SPE	Stock Price Estimation $\notin$ SF
SPP	Stock Price Prediction $\in$ SF
TF–IDF	Term Frequency-Inverse Document Frequency
TS	Text-to-Text Transfer Transformer
TSFM	Time Series Foundation Model
TM	Trend Matching
ULM	Unconditional Language Modeling
W2V	Word2Vec
XAI	Explainable AI

To maintain readability, the author refrains from itemizing every individual stock or asset in the list of abbreviations. The color-highlighted ticker symbols (e.g., BLK = BlackRock Inc.) allow for quick and straightforward identification via the exchanges listed in Chapter 4 when necessary.

Mathematical notations are explained in Section 6.2.

Further Acknowledgments

The stock data used for the models presented in this research was collected courtesy of research access kindly provided by Alpha Vantage[1].

Statement on the Use of Generative AI

This thesis used Generative AI (GAI) tools—specifically the ChatGPT-AI[2], the Claude-AI[3], and the DeepSeek-AI[4]—between 2023 and 2025 as assistive systems for writing refinement and software scaffolding. The author is solely responsible

[1] https://www.alphavantage.co

[2] https://chat.openai.com/

[3] https://claude.ai/

[4] https://www.deepseek.com/

for the scientific ideas, the experimental design, the data processing, the analysis and interpretation of results, and all conclusions. Model outputs were treated as drafts subject to verification; no confidential or proprietary data were uploaded to these systems.

All text and code generated with the assistance of GAI tools was critically reviewed, revised where necessary, and fully validated by the author. The author takes complete responsibility for the correctness, originality, and academic integrity of the final manuscript and experimental implementations. As GAI outputs may contain inaccuracies or unverifiable statements, they were used only as provisional drafts subject to systematic checking.

The role of GAI in the writing process was strictly limited to style improvements, shortening overly long passages, suggesting clearer structures, and maintaining consistent English grammar and technical terminology. No scientific thoughts, hypotheses, experimental setups, or references were generated by GAI. Literature research was carried out through web search and deep research functions of the GAIs; all references were read by the author and and the author did not rely on the GAI's outputs. Text passages were always provided beforehand by the author, at times in the form of bullet points or rough drafts, and GAI suggestions served only as language refinements.

The main use of GAI concerned programming support, primarily for debugging, generating helper or auxiliary functions, and suggesting more elegant implementations of existing code blocks using established libraries or idioms. Drafts produced by GAI were always tested against the author's own implementations to ensure correctness and consistency with the intended design. Tasks included writing small utilities (e.g., data loaders, preprocessing helpers, or reusable plotting functions), improving readability by modularizing repetitive code, and pointing out potential optimizations with standard PyTorch or NumPy functionalities.

GAI was also consulted for visualization code, such as preparing figures in `matplotlib`, creating canvases for embedding projections (e.g., PCA plots), or formatting result tables for clarity. These outputs were treated as scaffolding, with final plots and statistical analyses based entirely on the author's experimental data and manual interpretation. At no point did GAI design new models, invent training paradigms, define loss functions, or set hyperparameters. The conception of architectures (e.g., Stock2Vec, QMSE, recurrent and transformer-based variants, and pretraining objectives), the design of experimental protocols, and all methodological choices remain original contributions of the author.

In short, GAI was employed to accelerate routine programming tasks, to assist in debugging and visualization, and to polish the presentation of results, while the intellectual contributions—including the research framing, model design, training strategies, and empirical evaluation—are entirely the author's own.

Chapter 1

Introduction

This chapter introduces the topic, outlines the motivation, states the research questions, and situates the work within the literature.

This section is mainly based on the authors publications [220] and [224].

1.1 Motivation

Among all fields within ML, few have been as impressive in recent years as NLP. NLP is a subfield of artificial intelligence concerned with enabling computers to process, interpret, and generate human language. At its core, NLP focuses on sequential data, where meaning is derived not only from individual tokens (such as words) but also from their order and context. Advances in NLP have produced a variety of model architectures (e.g., recurrent neural networks, attention-based transformers) and training paradigms (e.g., self-supervised learning, pretraining-finetuning) that are now widely regarded as central to modern ML. Prominent examples of such transformer-based language models include BERT [40], GPT-2 [187], Transformer-XL [31], T5 [188], and LLaMA [216]. BERT is an encoder-only model designed with MLM and NSP, capturing bidirectional context. GPT-2, in contrast, is a decoder-only model optimized for autoregressive text generation.

Transformer-XL extends the standard transformer architecture by introducing a recurrence mechanism that enables the processing of longer sequences. T5 adopts a unified sequence-to-sequence framework, formulating every NLP task as a text-to-text problem. Finally, LLaMA denotes a recent family of efficient foundation models that achieve competitive performance at moderate scale.

Ongoing research has made NLP a key focus in AI research and applications. NLP has developed many innovative approaches and concepts that could potentially benefit other domains. However, the cross-domain application of these advancements remains limited, which is particularly concerning given the substantial resources and manpower invested in NLP, as well as the wealth of knowledge generated in the field. In most instances, only isolated components or models have been adapted for use in other domains, rather than leveraging the full potential of the techniques developed. The application of NLP techniques across diverse fields is not an unexplored topic. Models originally developed in the NLP domain, such as transformer models [219], have become standard in various ML areas, ranging from time series analysis [261] to computer vision [52]. Additionally, pretraining methods, widely employed in NLP, have been adopted across broader domains, including vision [52], multimodal language-vision tasks [118], and audio processing, where the attention mechanism has gained significant popularity [172]. Moreover, GNNs, inspired by word-token embeddings like W2V, have further demonstrated the cross-domain applicability of NLP innovations [182].

Transferring NLP techniques to time-series data remains relatively underexplored. Given the temporal structure inherent in both language and time series, such a transfer seems promising. Yet, there is little research aimed at leveraging NLP methodologies for analyzing multivariate time series data. A multivariate time series is a collection of time-dependent variables observed simultaneously at each time step. In contrast to univariate time series, where only a single variable evolves over time, multivariate time series involve multiple interdependent dimensions. Stock markets produce multivariate time series, since a company's stock price depends not only on its own past performance but also on trends in other companies, sectors, and global indices. This interconnected and high-dimensional nature makes multivariate time series analysis particularly challenging, yet also conceptually similar to NLP, where the meaning of a sentence emerges from the interplay of multiple tokens in sequence. The application of these concepts to stock prices is focused on in this thesis, as they present a particularly challenging problem due to their volatile and non-stationary nature. While SF lags behind in academic research, this is not only due to the inherent complexity of the problem but also a result of the relative lack of attention it has received in (publicly available) research as for example pointed out in [74]. Exploring the use of advanced NLP techniques in this domain may help bridge the gap and provide new insights for tackling such difficult forecasting tasks.

Financial profitability is a central incentive for applying ML to SF. However, the use of ML extends beyond the aim of gaining a financial profit; it can also contribute to economic modeling and empirical analysis. This is exemplified by the methodologies discussed in [204]. Additionally, SF may hold practical value for governmental and regulatory bodies, as noted in [271], suggesting that its strategic application could support regulatory initiatives. This is particularly evident for the proposed options presented in Section 9.1 to use the developed models in market regulation and risk modeling. Moreover, as indicated in [265], targeted interventions enabled by SF could support early-warning and risk-monitoring tools for regulators.

Beyond purely academic considerations, the potential real-world applications of transferring NLP strategies to financial forecasting are substantial. Comparable approaches connect DL methods to use-cases in trading, risk, and market monitoring [285]. Improved SF models may not only support institutional investors in portfolio optimization but also provide value for regulatory bodies in maintaining market stability, for central banks in assessing systemic risk, and for individual investors navigating increasingly digitalized trading platforms. More broadly, accurate forecasting tools can support macroeconomic planning, corporate financial management, and risk mitigation strategies.

Stock Forecasting in Machine Learning There are numerous challenges associated with SF, which will be addressed in detail later. SF refers to the use of computational models to predict the future behavior of financial assets, typically expressed as time-indexed price series. Depending on the methodological approach, SF may focus on predicting discrete movements (e.g., upward, downward, or neutral trends - SMP), exact future price levels (SPP), or volatility patterns. Broadly, two main strategies for analyzing stock market data exist: quantitative [36] and fundamental analysis [225]. Quantitative analysis (or technical analysis [286]) aims to predict future stock prices based on historic ones [36] and is usually guided by mathematical, statistical, or ML models and often relies on algorithmic execution at machine speed [82]. Fundamental analysis incorporates broader economic indicators, company performance, and market sentiment as well as other information sources like annual reports, news articles or earning call transcripts to name a few. As explained further in Section 2.3, the distinction between these two categories can be difficult in practice. If stock time series data is considered as an auxiliary problem that could benefit from techniques in NLP, it makes sense to first focus on time series analysis. Time series data is well suited to models that have proven effective in NLP because of their sequential structure. Meanwhile, the additional information used in fundamental analysis can be treated as another modality, similar to how multimodal models integrate different types of data in NLP.

While fundamental analysis is widely used in practice, this thesis focuses on quantitative methods, as SF is treated as an auxiliary problem that can benefit from NLP-inspired techniques. Quantitative analysis, which relies on historical price data, is supported by extensive research. Researchers argue that historical prices are either highly significant [197] [214] or the single most crucial factor in forecasting future trends [25].

Are Stocks Predictable? In the discipline of economics, various theoretical frameworks debate the extent to which future stock prices can be predicted, and a brief examination will be given of how recent research in ML approaches this issue.

The RWT, proposed by Kendall and Hill [107], suggests that because stock prices are inherently random, accurate prediction is impossible. The EMH asserts that all information available to investors is fully reflected in stock prices [60], with some scholars seeing a causal link between the EMH and the principles underlying the RWT [76]. Notably, certain researchers view the EMH as a concept within which stock prices may be forecasted, but within the bounds of market efficiency [138].

The EMH can be categorized into distinct gradations, ranging from the weak form, which posits that all historical price information is fully reflected in current asset prices, thereby rendering technical analysis ineffective, to the strong form, which asserts that all information, both public and private (including insider knowledge), is entirely incorporated into market prices, thus making it impossible for any investor, regardless of access to non-public information, to consistently achieve exceptional returns. This classification is addressed in the ML models constructed in [25], where the authors argue that certain investors frequently possess access to more information than the general public.

Many scholars critique the EMH, arguing that markets are not fully efficient, as highlighted in works such as [230] [134] or by promising results achieved through ML methods as in [204]. Xu et al. [247], for instance, contest the EMH referring to [148] by asserting that new information requires time to be fully incorporated into stock prices. Liu et al. [139] further support this stance by pointing to anomalies like the Post Earnings Announcement Drift and the effect of overlapping topics in earnings calls, which reveal intricate stock relationships and improve predictive models. The foundation of the ML-based quantitative approach is succinctly captured by Wang et al., who state that ‘Historical stock prices have proven to be strong indicators of future stock trends and are widely referenced in financial literature’ [234].

Mapping of NLP and Stock Forecasting In the following sections, the motivation behind the authors approach of testing the applicability of NLP techniques for multivariate time series prediction is presented, using SF as a representative example. The conceptual alignment and mutual benefits between NLP and SF methodologies are intended to be explored.

To highlight the suitability of the proposed approach, the symbol ⇔ is used to represent the two-way relationship between NLP and SF.

Specifically, the validity of

$$\text{NLP} \Leftrightarrow \text{SF} \equiv (\text{NLP} \Leftarrow \text{SF}) \land (\text{NLP} \Rightarrow \text{SF}) \tag{1.1}$$

is aimed to be demonstrated by showing that stock data shares conceptual and structural similarities with NLP data and by showing that SF challenges can be tackled by NLP strategies.

NLP Shares Conceptual and Data Characteristics with Stock Forecasting and Multivariate Time Series Prediction

$$\text{NLP} \xLeftarrow{\text{conceptual similarities}} \text{SF}$$

NLP and quantitative SF exhibit notable conceptual and data-driven similarities, highlighting their analytical convergence [220]. Both disciplines rely on sequential data patterns, where future values are inferred based on prior observations. As outlined in [234] and [220], the forecasting approach primarily leverages the capabilities of NLP models in general, and LLMs in particular, to predict the next token in a sequence — whereby tokens are interpreted as (historical) stock prices. In quantitative SF, historical price data is employed to project future stock movements, analogous to ULM in NLP, where predictions about forthcoming word tokens are made using only the initial part of a sentence [220].

While less commonly explored in SF, generative approaches that aim to predict entire price trends find a parallel in NLP tasks that autoregressively generate text from a given starting point [220]. These methods, though less prevalent, offer a broader view of potential future market movements akin to text generation in NLP. When contextual information, such as in fundamental analysis, is incorporated into stock price models, the process parallels CLM, which uses additional context for predictions [75].

Moreover, the structural similarities extend to data representation. In NLP, high-dimensional, concatenated word embeddings are used in language models to capture semantic and syntactic relationships and sequential dependencies within sentences. Similarly, stock price time series can be represented as temporally ordered market snapshots [220], with each snapshot containing multivariate data. These embeddings and time-ordered representations capture intricate dependencies within the data.

Furthermore, NLP offers an intriguing framework through the sequential arrangement of word tokens, which express their internal relationships via vector space representations and are ordered by sentence positions.

NLP Strategies can Tackle Stock Forecasting Challenges

$$\text{NLP} \xRightarrow{\text{tackles challenges}} \text{SF}$$

The use of NLP-based strategies for time series analysis is motivated at various points throughout the literature. For instance, [103] emphasize that ‘we encourage researchers and practitioners to recognize the potential of LLMs in advancing time series analysis and emphasize the need for trust in these related efforts’ [103] and further assert that ‘our standpoint is that LLMs can act as the central hub for understanding and advancing the analysis of time series data’ [103]. Similarly, [164] identify the application of LLMs for time series forecasting as a promising direction for future research.

Predicting stock prices is generally acknowledged as a challenging problem [286] [230] [231] [12] [168] [179] [124] and exact prices are often considered unpredictable [85]. [286] as a more introductory publication illustrates these challenges with simple time series techniques. A comprehensive synthesis of these challenges—non-stationarity, low signal-to-noise ratios, and evaluation pitfalls—can be found in [285]. Nevertheless SF has garnered the interest of researchers across various fields, particularly in ML.

As pointed out in [19] and [260] SF went from econometric time series techniques over ML to DL [285] [286]. Intuitively, DL is well-suited for stock price prediction, primarily because a fundamental capability of ML involves uncovering latent patterns within raw data, in this instance, time series stock data [14] and learning representations from the data. One of the key reasons for this focus is the potential of ML and DL technologies to address these challenges without the need for costly domain-specific expertise [12] [124] [231] [82].

Furthermore, these models offer the advantage of operating without the predefined assumptions [272] [124] [204] that typically bias (wrong) human analyses, which are notably prevalent in the financial sector. In addition, it was argued in [179] that the possibility of adapting the ML models to automated trading algorithms can lead to significantly faster response times. This also includes the challenge of unfavorable interactions between correctly predicted technical indicators and the need to balance them against each other, as discussed in [82].

Additionally, while traditional statistical models such as ARIMA and GARCH have been utilized in the past, they are often limited by inherent assumptions about linearity and stationarity [245] [272] [207], that may not be valid for stock market data [259] [286]. These models also lack the expressive power required to effectively model the complexities of the stock markets. Predicting stock prices remains a highly challenging task, largely due to the inherent complexity and dynamic nature of financial markets. A comprehensive review of the existing literature, as detailed in Chapter 2, reveals the manifold key reasons for the difficulty in stock price prediction which are summarized in the following:

One of the foremost challenges is the non-stationary nature of stock data [12] [242] [230] [25] [209] [246] [264] [168] [231]. Financial markets are characterized by constant distribution shifts [97] [230] [242] [264] [168], making traditional prediction methods less effective. This data also exhibits a low signal-to-noise ratio [12] [33] [71] [179] [214], making it hard to extract meaningful trends from the stochastic behavior of stock prices [214] [259] [180]. Furthermore, stock data is not independently and identically distributed (i.i.d.) [230] [245] [134], an assumption upon which many ML strategies are based. Homogeneity within the dataset represents a significant challenge, as discussed by Gao et al. [71]. As explained by Gao et al., this phenomenon occurs when related stocks (for example, from similar industries) show similar behavior. This homogeneity reduces the number of stocks with distinct informational characteristics, limiting the discriminative features available for analysis.

To mitigate these challenges, researchers have adopted a variety of approaches to gain some predictive power over stock price movements. As Fan and Shen highlight in their work [62], SF models frequently seek to exploit three key sources of predictive insight: 1) temporal correlations, 2) indicator correlations, and 3) (delayed) inter-stock correlations. These approaches are typically complemented by analyzing seasonal trends [214] [28] [201] [20] and aligning predictions with global market movements and overall market correlations / trends [62] [124] [204] [240]. The latter approach, of course, represents a specific form of inter-stock correlation identification. It is argued that the domain of NLP offers innovative solutions to the challenges faced by current approaches in stock market analysis and predictive modeling.

One key aspect is the use of indicator correlations, where weighting different features and constructing higher-level representations is a fundamental ability of ML and DL approaches. This technique is prevalent in most proposed models, and its importance in capturing complex feature interactions is well-established.

Another critical area are the inter-stock correlations. One of the most popular techniques in NLP, word embeddings, encodes word-tokens as high-dimensional vectors to represent their semantic relationships. This approach offers a practical way to model stock correlations, going beyond simple point-wise adjacency matrices and capturing more complex relationships between stocks.

Temporal correlations represent another intrinsic characteristic of NLP models. The ability to set word-tokens or speech concepts in their correct temporal order—whether word, sentence, or phrase order—is essential for deriving a high-level understanding of textual data. Similarly, temporal relationships in stock market data are crucial for understanding market dynamics over time.

Furthermore, spatio-temporal processing is a key feature of many modern SF models. This approach connects the multivariate and intercorrelated nature of stock data with the ability to integrate temporal dimensions. This mirrors the functionality of advanced NLP models, such as LLMs, which process word-token embeddings that reflect their interrelationships trough vector space positions and temporal context by arranging them in the order they appear in a sentence. This temporal arrangement can also be adapted to stock models. Adapting LLMs directly for SF is therefore considered a worthwhile research direction.

Seasonal trends in financial data can be understood as recurring patterns that manifest with varying granularity's. Notable examples include phenomena from 'calendar anomalies' [99] like the 'January effect or the turn-of-the-year' [240] down to weekly ones such as stocks having higher movements on Friday [240], weekday-effects or the seasonal profitability of certain industries [73]. Multiple timeframe analysis (Multiscale Analysis) for stocks is a trading technique where analysts or traders evaluate the same stock across different time intervals (e.g., daily, weekly, monthly) to identify trends and align long-term and short-term market movements for more informed decision-making[1] [273]. The challenge of processing time series data across different frequencies bears similarities to the general hierarchical structure observed in language. Language, in its essence, is hierarchically organized, from the smallest units such as letters, to syllables, words, parts of speech, sentences, paragraphs, chapters, and ultimately entire texts

[1] https://www.tradingview.com/education/mtfa/

[220]. This hierarchical structure offers a compelling analogy for the processing of financial time series data. Specifically, a hierarchical and frequency-sensitive approach to language processing could be suitably adapted to the analysis of stock market data.

Global correlations and market sentiments are another important consideration [204] [124] [240]. These can be attributed to general stock correlation expressions, but they also require placing currently processed data within an appropriate context. In NLP, document-level analysis, such as that employed in Doc2Vec models, incorporates additional contextual information (e.g. document type, author, or title) to enhance the model's understanding. Similarly, integrating contextual data into stock market analysis will provide models with a deeper understanding of market dynamics. The use of contextual information is a key component of CLM [75], a widely used strategy in NLP. This approach will be particularly useful for modeling stock market data, where context—such as sector performance, geopolitical events, or economic indicators—plays a significant role. Moreover, adapting the NSP task from BERT [40] offers a promising avenue for training models to develop a comprehensive semantic understanding (in contrast to MLM used to train the syntactic understanding). In the context of stock market analysis, this adaptation might help the model grasp overall market dynamics and processes, creating a more robust and nuanced understanding of temporal trends and correlations within the financial landscape.

In addition to approaches aimed at predicting stock prices, it is crucial to address the common challenges associated with stock time series data. It is argued that NLP techniques can be leveraged to overcome these issues as well.

Surprisingly, given that the stock market is a human-created system, there is a notable lack of publicly available stock data [264] [184] [71] [85] [84] [163]. For interday data or coarser granularities, the scarcity of time steps can be attributed to the relatively short history of stock market records. Furthermore, the limited availability of long-term stock price records is particularly evident for smaller companies. This shortage of data presents a significant challenge, especially when attempting to model specific companies or markets with finetuned predictive models. While NLP tasks generally benefit from abundant datasets, certain specialized tasks—such as finetuning or human-annotated tasks—can also face data limitations. Modern NLP techniques, however, provide two key strategies to mitigate these challenges. First, models can be pretrained on large datasets for generalized language understanding and subsequently finetuned on smaller, task-specific datasets. This same approach could be applied to stock data, where models could be pretrained on long-term stock data of well-established companies with the same temporal granularities. Alternatively, pretraining on finer granularity data can serve as a first step for learning broader stock market dynamics. Finetuning can than be used for predicting stock trends for newly emerged or publicly listed companies, which may only have limited historical stock data, as well as for companies operating in niche markets where data scarcity is a common issue. Secondly, in NLP, self-supervised tasks such as MLM and NSP (explained later in this section) are designed to allow the same sentence to be processed multiple times, each instance representing a distinct data point. Adapting this technique for stock data will similarly generate additional training instances from the same stock record, enhancing model robustness even with limited data availability.

Distribution shifts pose a significant challenge in stock market analysis due to the ever-changing, non-stationary nature of stock data [242]. Market sentiments can shift rapidly, rendering previously learned patterns and rules obsolete. Addressing these shifts requires models that can adapt quickly to new information and changing conditions.

NLP offers a promising conceptual solution to this problem through the increasingly popular approaches of few-shot and zero-shot learning [264]. Few-shot learning allows models to perform tasks with only a small number of training examples [9], while zero-shot learning enables models to handle tasks without any task-specific examples by leveraging prior knowledge from related tasks [243].

Applied to the stock market, these techniques suggest that models can be trained to quickly adapt to new and rapidly changing market conditions. Few-shot learning could help a model adjust its predictions after being exposed to just a few examples from a new distribution. Similarly, zero-shot learning could enable the model to apply pre-existing knowledge from related financial patterns or markets to navigate novel situations. Given the fast-paced nature of distribution shifts in financial markets, the ability of a model to adapt swiftly and effectively after minimal exposure to new data is highly advantageous.

While non-stationarity, stochasticity, and low signal-to-noise ratios are present challenges in stock market analysis, these issues typically arise only under exceptional circumstances or niche research areas within the NLP domain. For example, spelling mistakes or low-quality textual inputs can pose difficulties for NLP models, but these represent a relatively specialized subset of the broader NLP field.

The author aims to explore the use of models capable of processing longer sequences, addressing a gap in the current literature where many SF models primarily focus on short sequences (see Section 2.3). Handling long sequences in NLP presents significant challenges. This is primarily due to the limitations of most modern NLP models, particularly LLMs based on transformers, which suffer from quadratic time and space complexity. Longer input windows may help models contextualize non-stationary behavior. By processing longer sequences, the model is expected to develop a deeper understanding of the underlying temporal dynamics and achieve a meta-awareness of non-stationary patterns. A conceptually similar approach is explored in [65], where artificial noise is introduced into the model's latent representations. This method aims to familiarize the model with stochastic variations, thereby enhancing its robustness to uncertainty. Such an approach may also be viewed as fostering a form of meta-understanding within the model. Short sequences may be insufficient for capturing such complexities, as they only provide inputs with changing statistical characteristics without offering insight into how the time series evolved to these points / these changing characteristics. Furthermore, longer sequences may enhance the model's ability to contextualize data, allowing for the identification of long-term trends and dynamics that would otherwise be obscured as noise in shorter inputs. In the realm of technical analysis for stock charts, specifically within the indicator-based manual chart analysis, numerous indicators are employed to discern coarse trends despite limited lookback periods. Among these, moving average indicators are prevalent. Additionally, as outlined in [133], the RSI [286] serves as a mechanism to identify coarse-grained trends. This ability is key to reducing the risk of poor decisions caused by overly narrow analytical perspectives — something that models handling longer sequences are also better equipped to address.

The pretraining approaches offer a promising avenue for addressing several challenges in SF. In particular, MLM plays a crucial role in this framework. MLM is a technique in NLP where certain words in a sentence are hidden or 'masked', and the model's task is to predict these masked words based on the surrounding context [40]. MLM is a form of CLM where the model is tasked with predicting missing words in a sequence based on the surrounding words as context. When this concept is adapted to financial markets, the model is similarly challenged to predict the next stock price based on historical price sequences, while incorporating future price data during pretraining.

Although future prices are unavailable in practical applications, providing them as conditional context during pretraining enables the model to capture underlying price dynamics that would otherwise be interpreted as noise. By utilizing additional contextual information, the model can better explain underlying dynamics that would remain undiscovered in a traditional predictive task.

Additionally, the NSP adaptation addresses the inherent stochasticity in financial data. By training the model to distinguish between stock trends that are connected and those that are not, an NSP adaptation is hoped to help the model focus on trends it can reliably predict, while acknowledging that certain developments—driven by unforeseen events—will manifest as stochastic and unpredictable.

Moreover, the application of CLM and the incorporation of techniques like Doc2Vec further enhance the model's ability to handle non-stationarity in market data. By providing additional information on the current state of the market, these methods are expected to improve the model's adaptability to changing market conditions, hopefully enabling it to respond more effectively to shifts in market dynamics.

Although not a typical issue in the domain of SF, having pretrained, general stock models proves advantageous, as it eliminates the need for training models from scratch with every downstream task.

1.2 Research Aims and Questions

This thesis aims to explore whether and how techniques from NLP can be systematically adapted to the domain of quantitative multivariate stock price processing/SF, with the long-term objective of developing domain-specific foundation models for financial time series. During this research, and in light of recent developments in the wider research community, the following reserach questions have emerged.

Research Questions

- **Q1:** Which Strategies from the NLP Area can be Adapted for Quantitative Multivariate Stock Price Data and How Can We Use Them?
- **Q2:** To What Extent can Adapted Strategies Contribute to Improving Prediction?
- **Q3:** How Can Effective Foundation Models for Quantitative Stock Data be Built?

Figure 1.1: Research Questions.

The initial proposal of the research questions was made in [220].

1.3 Thesis Outline

The remainder of the thesis is organized as follows. First, a comprehensive review of the relevant literature is conducted with three primary objectives: i) to briefly survey existing NLP models and ii) approaches relevant to financial forecasting and iii) to identify and position this thesis within gaps in the current literature (thereby justifying the novelty of the research and clarifying which established aspects no longer require extensive examination).

Having established the background and gaps in existing knowledge, the discussion turns to (transformer based) baseline approaches. Transformers serve as the primary baseline, reflecting their central role in modern NLP. In addition, recurrent architectures such as RNNs and LSTMs are included as comparative baselines, reflecting their established role as quasi-standard models in sequence forecasting, as documented in prior work (e.g., [280] [5]). This convention is further supported by studies such as [286], which mentions these models as standard reference points. A detailed rationale for their inclusion in the experimental design is provided in Section 6.5.

These serve as crucial reference points, allowing for objective assessment of newly proposed methods, especially in view of the lack of standardized datasets. The baselines provide a clear framework for comparison, ensuring that improvements introduced by novel adaptations can be measured against well-understood metrics and models.

Next, the focus shifts to the specific strategies devised for adapting NLP techniques to financial time series data. Following the general framework of classical NLP, the first step involves adapting W2V models (S2V) to learn meaningful latent representations from financial data. Additionally, although Doc2Vec algorithms do not typically occupy a central role in standard NLP pipelines, they are examined (as QMSEs) for a potential utility in certain specialized scenarios, particularly those calling for richer contextual embeddings that may provide an advantage in some edge cases of SF.

Subsequent sections explore three major avenues of adaptation in more detail. First, the integration of pretraining procedures, including MLM and NSP adaptions, is investigated, Second, the possibility of incorporating hierarchical processing is evaluated to capture multi-level patterns in the data, an idea reflecting the multi-scale nature of both financial time series and the hierarchical structures seen in language. Third, the potential impact of extending sequence length is examined, building on current NLP research trends that focus on harnessing longer input sequences for enhanced predictive power.

On this foundation, the thesis examines the role of LLMs in unifying these strategies. Thanks to their extensive pretraining, hierarchical understanding, and compatibility with diverse embedding methods, adapted LLMs (ASMs) show strong potential for financial forecasting tasks. In support of this view, Wang provides an empirical evaluation of LLMs for asset return prediction, offering an early benchmark for the emerging LLM-for-returns literature [288]. Their built-in capacity for few-shot learning, integration of additional contextual information, and generalizability shows their relevance as potential SF model. The schematic structure of this procedure is shown in Figure 1.2.

The discussion then reviews the experimental results in relation to the original goals and hypotheses, highlighting where the outcomes matched expectations and examining any differences. This assessment includes a critical reflection on how effectively the adapted NLP strategies perform relative to the established baselines, as well as a broader commentary on the prospects and constraints of transferring linguistic models to the financial domain.

Finally, the conclusion brings together the main insights gained from applying NLP techniques to stock price data. In acknowledging both the strengths and limitations of the examined methods, it points toward the most promising directions for further exploration. Given the research results that some of the models may prove less suitable for SF than initially expected, future work will consider alternative applications of foundation models and articulate how the lessons learned from this research could guide subsequent developments in the interdisciplinary space between NLP and financial modeling specifically or time series modeling in general.

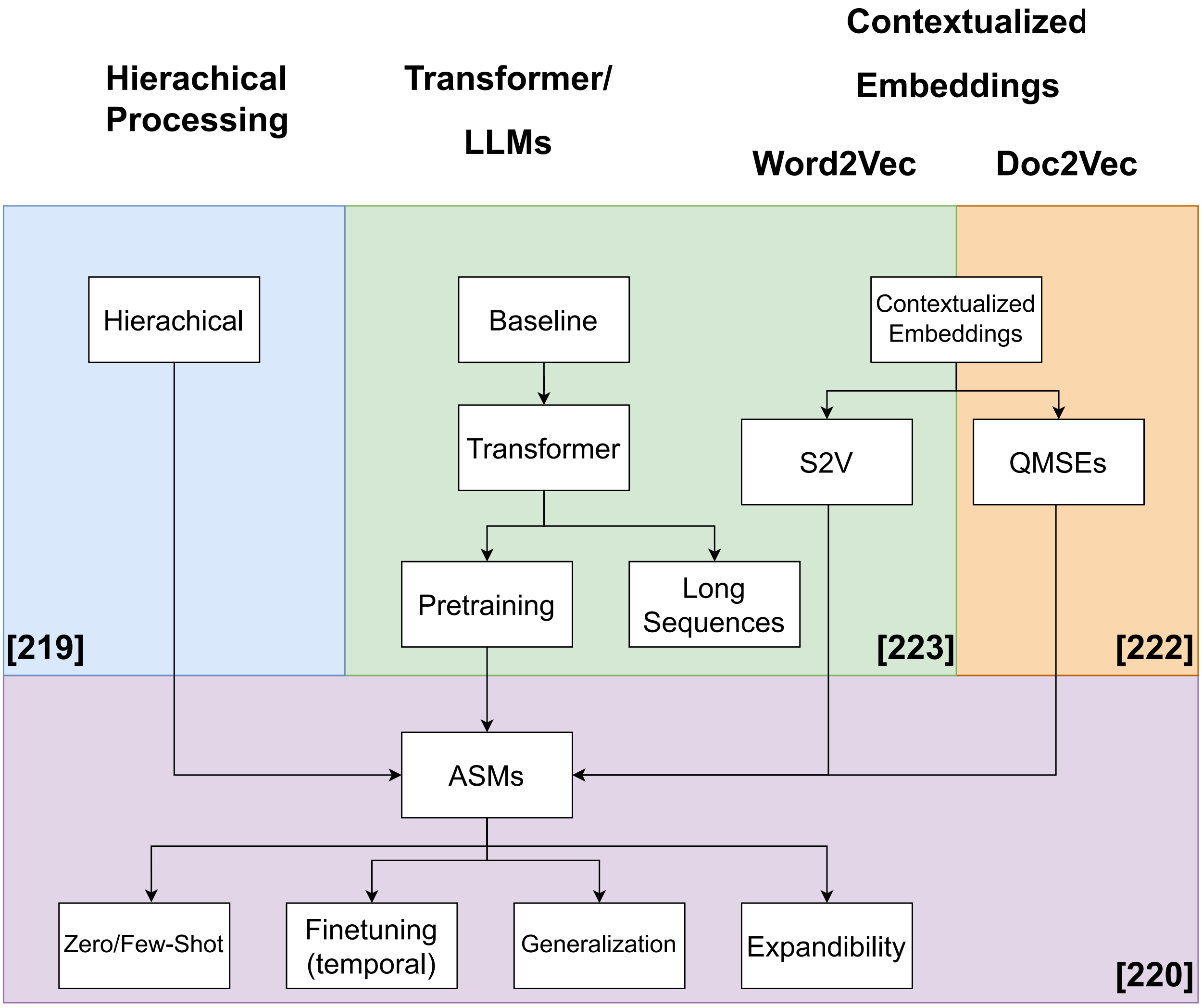


Figure 1.2: Thematic structure of the thesis from NLP concepts to the foundation model for quantitative stock time series.

Chapter 2

Literature Review

This review is split into three sections that match the thesis' two themes - NLP, quantitative SF and NLP inspired techniques for SF. The initial section provides a concise overview of the adapted NLP techniques. Given that the NLP methodologies are not significantly investigated by the proposed models discussed in this thesis, the exposition in this section is intentionally brief to maintain a focused scope of discussion.

The second section gives a review of typical quantitative SF approaches.

The third section surveys ML-based approaches with NLP relation/relation to the proposed approaches to SF in recent work. The systematic review encompasses a comprehensive categorization of relevant literature with a focus on proposed approaches related to NLP. The goal is to clarify which SF setups and methods exist and which patterns are relevant to the models in this thesis. The thesis engages with current research to highlight the specific gaps it aims to fill.

In reviewing related work, an extensive discussion of models based on common backbone architectures such as transformers, RNNs, and LSTMs is omitted. Since these models are widely used across many ML applications, covering them in detail would not highlight the unique contributions of this thesis. Instead, the focus is narrowed to models that exhibit conceptual parallels or adapt methodologies akin to those proposed in the present research, namely from the NLP domain, thus emphasizing the novelty.

2.1 Overview of Suitable NLP Approaches

In the subsequent discussion, concepts from the field of NLP that have been adapted for the present study will be explored. The relevance of these approaches has been established in Chapter 1. First, a more in-depth overview of modern NLP models is provided in the following.

A Guided Introduction to NLP The following section provides a more in-depth look at NLP techniques. This paragraph complements the concepts introduced in the rest of the section and aims to avoid repetition.

Classical (rather more distant) NLP represents texts with sparse vectors such as Bag-of-Words (BoW) and TF–IDF, while word n-gram language models estimate next-word probabilities under a Markov assumption and are evaluated via perplexity [313]. Smoothing (e.g., Kneser–Ney) is essential to handle data sparsity and improves generalization in practical settings [314].

HMMs provide probabilistic sequence models that were widely used for Part-of-Speech tagging - pairing simple first-order dependencies with efficient decoding [315]. Conditional Random Fields (CRFs) later became standard for labeling tasks such as Named Entity Recognition and chunking, directly modeling $\mathcal{P}(\mathcal{X} = \mathbf{y} \mid \mathbf{x})$ with rich features and global sequence consistency [316].

Syntactic parsing distinguishes constituency (phrase-structure) and dependency formalisms; both yield explicit structure beyond tokens and are useful for downstream extraction and reasoning [318]. Modern pipelines historically used fast transition-based dependency parsers, while chart-based algorithms remained the basis for exact inference in grammar-based systems [317].

Before neural models, strong baselines combined BoW/TF–IDF with linear classifiers (Naive Bayes, Logistic Regression) and margin-based Support Vector Machines, which remain competitive for small data and high interpretability [318] [319].

Modern NLP systems can be understood as a pipeline in which raw text is converted into numeric representations, enriched with context, and adapted to downstream tasks. To enter this pipeline, text is segmented into tokens by subword methods such as byte-pair encoding, which reduces out-of-vocabulary issues and stabilizes training across domains [290]. A widely used alternative is SentencePiece, which implements an independent unigram model and treats text as a stream of bytes, facilitating multilingual and domain-agnostic processing [291]. This also aligns with observations from neural arithmetic research that simple inductive biases can improve extrapolation on numbers [306].

After tokenization, tokens are mapped to vectors and contextualized by transformer layers. In older approaches, these embeddings were pre-trained using Word2Vec techniques (cf. Section 2.1). Transformers compute contextualized representations via self-attention, allowing each token to integrate information from all other tokens in the sequence (cf. Section 2.1). In encoder–decoder configurations the encoder forms bidirectional context while the decoder generates autoregressively, whereas large language models in practice are often implemented as decoder-only stacks. Compared to static embeddings such as those obtained with Word2Vec, transformer layers yield dynamic token representations that are conditioned on the entire input, which is central to the transfer discussed in Section 2.1. During pretraining, different objectives shape the emergent capabilities; span corruption in a text-to-text setup has been shown to strengthen multi-token reasoning and transfer [188]. Denoising with deletion, infilling, and sentence permutation supports both comprehension and generation in encoder–decoder models [297]. Permutation-based autoregressive training retains causal generation while exposing the model to bidirectional signal at training time [298]. Scaling recipes further improve effectiveness through larger batches, longer training, and dynamic masking without changing the core architecture [299]. Fine-tuning (see Section 2.1) attaches a task-specific head and adapts parameters end-to-end when sufficient data and compute are available, while parameter-efficient strategies reduce the trainable footprint by design. Adapter layers introduce small bottlenecks inside a frozen backbone to enable efficient task adaptation and multi-task reuse [300]. Low-rank adaptation factorizes weight updates to approximate full fine-tuning quality with orders of magnitude fewer trainable parameters [301]. Prefix tuning optimizes short continuous prefixes that steer the transformer without modifying the backbone weights [302]. Prompt tuning learns compact soft prompts that are especially effective when label budgets are small and model scales are large [303]. Positional information is essential because self-attention is permutation-invariant. Relative position representations are frequently employed to improve locality and compositionality and to avoid the fixed-length limitations of absolute encodings [205]. Rotary position embeddings introduce a complex rotation in embedding space and have been shown to improve length extrapolation without architectural changes [292]. Attention with linear biases enables models trained on short inputs to generalize to longer ones by incorporating a distance-dependent bias at inference time [293]. These aspects are relevant for the $F^{\langle \mathrm{T} \rangle}$ and ASMs models introduced in Section 2.1.

Scaling to long contexts is both a modeling and a systems problem as already mentioned in Chapter 1. Linearized attention approximates softmax attention to bring memory use closer to linear in sequence length while maintaining competitive accuracy [294]. In parallel, fused kernels such as FlashAttention compute exact attention with IO-aware tiling, yielding substantial training and inference speedups without approximation error [295]. In practice, long-context robustness emerges from combining length-friendly positional encodings with either linear attention or fused kernels, while recurrent formulations from Section 2.1 are reserved for very long sequences.

Keeping knowledge current requires external grounding. Retrieval-augmented generation augments a generator with a non-parametric memory that can be refreshed without retraining, mitigating concept drift in fast-moving domains [303]. Dense passage retrieval provides the retrieval side through dual encoders and a vector index, enabling efficient look-up of relevant passages at inference time [304]. For financial text, this design supplies provenance and timeliness that are difficult to obtain from parametric memory alone. In the context of this thesis, S2V embeddings are considered as a basis for comparing and retrieving similar stocks (see Section 3.0.2).

Domain specialization illustrates the trade-off between breadth and depth. Large mixed-domain models with a strong financial slice have demonstrated improved performance on finance-focused tasks while maintaining general capabilities [306]. Earlier domain-adaptive work on financial sentiment showed that moderate, targeted adaptation already yields notable gains on classification benchmarks [307]. Throughout this thesis, the models are treated as modular components; their combination with textual data is primarily outlined as future work in Section 9.2, and the demonstrated Doc2Vec integration illustrates additional numerical modularities.

Word2Vec Most NLP models, including LLMs, rely on word vector embeddings. These embeddings encode individual word tokens by their position within a vector space, thereby capturing the relational semantics inherent to language. Such embeddings are typically pretrained to encapsulate general language features, which can then be employed directly in downstream NLP tasks or further refined during task-specific training. Noteworthy among the algorithms that facilitate this are the SG and CBOW models [156]. The SG algorithm aims to predict the contextual words surrounding a target word, whereas the CBOW model predicts a target word based on its context [156].

In the domain of NLP, textual data is generally deconstructed into discrete word tokens, represented as $\tilde{v}^{(t)} \in \tilde{V} \subseteq \mathbb{N}$. Each $\tilde{v}^{(t)}$ is associated with a corresponding embedding vector $\tilde{e}^{(t)} \in \mathbb{R}^{\tilde{\xi}}$. These embeddings are trainable and play a crucial role in the computational efficacy of various models.

In Word2Vec and some LLM pretraining phases, the embedding $\tilde{e}$ is trained to capture the semantics of its token. As a result, embeddings of semantically related words tend to be close in the embedding space. Furthermore, it typically enables the geometric relationship between embedding pairs to reflect the semantic relationships between their respective words. For example, the relationship between the embeddings $\tilde{e}_{\text{king}}$ and $\tilde{e}_{\text{queen}}$ for the words ‘king’ and ‘queen’ mirrors the relationship between $\tilde{e}_{\text{women}}$ and $\tilde{e}_{\text{women}}$ for the words ‘man’ and ‘woman’ respectively, in terms of both angular and distance metrics within the embedding space.

Pretraining / Large Language Models As described in [222] and Section 2.1, LLMs follow three stages. In the initial phase, input text undergoes tokenization, during which each different word-token from a sequence length $\tilde{l}$ is assigned an index from a predefined vocabulary $\tilde{V} \subset \mathbb{N}$. Following this, the subsequent phase involves the generation of contextualized embedding tensors for each index. This embedding, potentially utilizing pretrained models, leverages techniques such as the W2V methodology, as discussed in the preceding section. The final phase involves the concatenation of the individual embeddings into a bi-axial tensor, which then serves as the input for further processing by the model.

LLMs are typically pretrained. Pretraining provides a general language model before any task-specific fine-tuning [220]. This training enables the models to pick up on subtle word meanings, syntactic patterns, and broader language structures, making them more effective across a range of language tasks. Pretraining involves exposing a model to a voluminous corpus of textual data, often sourced from multiple heterogeneous sources. The process predominantly employs self-supervised learning mechanisms, one of which is MLM [40]. In MLM, certain tokens in the text are replaced with a special [MLM] token, prompting the model to predict the hidden word using the unmasked context words. NSP trains the model to determine whether one sentence logically follows another, enhancing its ability to understand coherent text [40].

Fine-tuning adapts the pretrained model to a specific task or domain. This stage uses supervised learning with labeled data to adapt the model's general language understanding to specific tasks, improving its performance in applications such as translation, summarization, or question answering.

LLMs exhibit substantial domain generalization capabilities. This attribute enables them to transfer and apply their acquired knowledge across a wide spectrum of subjects and sectors without necessitating explicit training for each specific domain. Furthermore, approaches such as zero-shot [243] and few-shot learning [9] underscore the LLMs' remarkable ability to generalize effectively. In [67], it has already been mentioned that zero and few shot abilities of LLMs can be of interest for time series models.

Hierarchical models As discussed in Chapter 1, language exhibits a hierarchical structure to a certain extent [220]. Similarly, financial markets are characterized by various cyclical patterns, including trading behaviors [82], cash flow timings [82], and interest rate fluctuations, which manifest across multiple temporal scales and are inherently hierarchical in nature.

This hierarchical characteristic has motivated the development of financial models that process inputs across different frequencies, as outlined in Section 3.0.3.

Notably, such ideas have been explored beyond quantitative stock market data to domains such as audio processing and, most relevant to this thesis, NLP. Among these hierarchical models, transformer architectures have gained prominence due to their ability to effectively capture structured dependencies in sequential data, including audio signals [250]. Moreover, the framework presented in [279] underscores the advantages of token-level processing over sentence-level approaches. In addition, the FAST model [201] integrates time-aware LSTM networks to address the non-uniform temporal distribution of textual data throughout the trading day. The study by Koutník et al. [112] focused on spoken word classification, where the proposed Clockwork RNN architecture proved effective in modeling temporal dependencies by partitioning the hidden layer into modules operating at different clock rates. This design enables the network to integrate both fine-grained and long-range temporal information. In the context of this thesis, these properties are particularly valuable, as financial time series data—similar to language and speech—exhibit hierarchical and multi-scale temporal structures. Consequently, Clockwork RNNs provide a principled approach for capturing such dynamics in market-related prediction tasks.

Recurrent Transformer Transformers encounter intrinsic constraints related to the time complexity of $\mathcal{O}(n^2 \cdot \xi)$ and space complexity of $\mathcal{O}(n^2 + n \cdot \xi)$ [219]. Such limitations are notably acute within the domain of NLP, where processing extensive texts remains a challenge. As a result, handling multi-page or multi-document text has become a key topic.

Recent methodologies extend beyond traditional attention mechanisms, such as global or local attention [7]. A practical direction is to process long texts in segments with recurrent state.

Recurrent transformers integrate a recurrent architecture within the transformer, thereby enabling the model to incrementally process segments of the sequence. At each iteration, the model refines its hidden states by incorporating information from the current segment alongside previously accumulated states, thereby effectively capturing long-term dependencies while optimizing computational efficiency. This hybrid combines local parallel attention with recurrent state for long-range dependencies or vice versa. Representative examples of recurrent transformer architectures include the ones mentioned in [224]; TransformerXL [31], Recurrent Memory Transformer (RMT) [10], and Block Recurrent Transformer [94].

Doc2Vec In Chapter 1, it was hypothesized that the generation of summaries from market data could potentially augment the informational input to the model. However, the task of generating summaries in NLP presents a challenge. These difficulties primarily arise from the need for extensive and costly datasets. Moreover, the inherent difficulty of evaluating the quality of generated summaries adds further complexity to the assessment process [39].

Doc2Vec provides document-level embeddings for downstream analysis. Unlike conventional (W2V) models that merely embed individual word tokens, Doc2Vec extends this capability to encompass more extensive textual units, including full sentences and entire documents [114] [109]. It produces embeddings for sentences and documents that can be seen as an abstract summary of the text.

These resultant vectorized representations encapsulate texts, capturing not only surface-level elements but also the deeper semantic content inherent within them. Such embeddings facilitate numerous applications, including document retrieval, by enabling the analysis and comparison of extended textual similarities.

Furthermore, Doc2Vec has been applied within the financial sector. For instance, [2] illustrates how embedding news paragraphs with Doc2Vec captures the evolving narratives within financial markets. Another research [55] investigates the application of Doc2Vec for event embeddings at both sentence and document levels.

2.2 Non-NLP baselines for financial forecasting

Beyond language-centric approaches, a broad spectrum of ‘price–only’ and market-structure methods establish essential baselines and complementary perspectives for equity forecasting. At one end stand classical econometric views that treat prices (or returns) as close to a martingale difference sequence, implying that naive or low-dimensional linear predictors provide strong yardsticks. Early empirical work documented price changes that are, to a first approximation, serially uncorrelated, foreshadowing the random-walk benchmark [107]. The weak-form EMH sharpened that idea: conditional on the information set embedded in past prices, risk-adjusted excess returns should be unpredictable [60]. In practice, these perspectives justify simple—but informative—baselines such as random-walk, historical mean of returns, and rolling linear autoregressions on (stationarized) returns. Even when more sophisticated models are deployed, such baselines remain critical to detect overfitting and to quantify economically meaningful gains. Portfolio textbooks codify this attitude by emphasizing that model performance ultimately must be judged in a risk–return framework rather than by point-forecast accuracy alone [36].

Price-only deep learning starts from recurrent architectures and their refinements. LSTM networks address vanishing gradients and capture medium-horizon dependencies [86]. Early applications to equities demonstrate feasibility on daily data [21] [41], with later work exploring encoder–decoder variants and GRUs across stocks and even crypto [46]. Convolutional front-ends feeding bidirectional LSTMs offer a lightweight alternative that aggregates local patterns before temporal integration [56]. Attention mechanisms then generalize these ideas: LSTM-associated network models leverage learned relevance over input timesteps [42], while hierarchical or multi-scale transformers aim to reconcile multiple temporal granularities [43]. More recently, domain-tailored transformers and MLP mixers have shown that leaner sequence mixers can match or exceed heavier attention stacks when appropriately regularized and windowed [74], [62]. Crucially, empirical performance is sensitive to the lookback window and forecast horizon; ensemble strategies across multiple windows mitigate this sensitivity and stabilize realized returns [203].

Forecasts become economically actionable only after mapping to risk. In the classical mean–variance paradigm, signals are evaluated by their contribution to portfolio-level moments—expected return and variance—and by risk-adjusted metrics (e.g., Sharpe), not solely by directional accuracy or RMSE [153] [36]. From a market-dynamics angle, volatility itself is a forecasting target with direct portfolio implications. High-frequency estimators of integrated volatility must explicitly correct for market microstructure noise; failure to do so can contaminate daily risk forecasts and backtests [266]. Recent ML studies explore the continuum from GARCH-type baselines to neural volatility forecasters, reporting tangible gains in realized-volatility prediction when architectures incorporate temporal weighting or nonlinear feature interactions [270]. On the allocation side, representation learning has been used to compress cross-sections into low-dimensional factors that improve pricing and risk control [81]. RL reframes portfolio choice as sequential decision-making: policies are trained to trade off expected return and risk, with recent work emphasizing data efficiency and transaction-cost awareness [100]. These strands are complementary: better volatility forecasts stabilize risk budgets, while learned portfolio policies translate predictive structure into implementable weight trajectories.

At intraday horizons, microstructure effects dominate, reshaping both inputs and targets. Observed transaction prices and quotes are contaminated by discrete tick sizes, bid–ask bounce, and asynchronous trading, so microstructure-robust estimators are required even to recover integrated volatility from high-frequency data [266]. Beyond noise correction, predictive content often resides in multi-frequency patterns—e.g., the interaction of short-cycle order-flow bursts with longer-cycle trends—which motivates models that explicitly decompose or attend across temporal scales [267]. In LOB-based settings, signals include queue imbalances, depth dynamics, and cancellation/arrival intensities; while many equity DL studies center on daily bars, pre-trained financial models increasingly leverage fine-grained price–volume structure for price-movement forecasting, blurring the line between ‘price-only’ and microstructure-aware inputs [61]. In all such setups, proper evaluation must account for latency, fill probabilities, and transaction costs, since edge at the quote level can evaporate after execution frictions. Methodologically, this strand connects naturally to the DL architectures above: attention over multi-scale features or hierarchical temporal encoders is a practical way to fuse LOB-state dynamics with lower-frequency context.

2.3 Review of the Current Stock Forecasting Research

In the subsequent section, methodologies within the SF domain that can be linked to the adapted NLP strategies are explored. In doing so, the relevance of the proposed methodologies is justified, and the original contribution of this work to the field is highlighted.

Review System SF in ML is a large and fast-growing area. Given the sheer volume of publications, it is impossible to address each one comprehensively within the confines of this thesis. A focused review is therefore needed to select relevant publications. For a publication to be considered relevant, it must not only utilize methodologies stemming from recent breakthroughs in NLP but also offer solutions and motivations that address the issues outlined in Chapter 1.

The development of this review system is inspired by methodologies employed in contemporary literature reviews. Specifically, the framework proposed in [282] has significantly influenced this thesis review system's design, providing an approach for filtering and evaluating relevant literature efficiently and effectively.

In this study, the search methodology delineated in [282] is adopted to bridge the research publication gap from the issuance of [282] until the conclusion of this thesis. The keyword search mechanism outlined therein is endeavored to be replicated to the fullest extent practicable (see Appendix A.7).

The scope of the research is expanded to include works published subsequent to December 2022, thus extending beyond the last reference date mentioned in [282], which cited [281] as the most recent publication.

Given the primary focus on the adaptation of NLP strategies, a set of keywords has been empirically derived from pertinent publications. The augmented keywords are as follows:

- Pre-trained / Pretrained (in combination with the other keywords)
- Finetuned / Fine-tuned (in combination with the other keywords)

as well as

- Stock Embeddings
- Finance Embeddings
- Financial Embeddings
- Foundation Models
- Recurrent Transformer

Details of the review system are given in [282]. Details regarding the modifications implemented are elaborated in Section Appendix A.7.

Additionally, the literature review encompasses publications that are relevant to the thesis for various reasons. These include papers explicitly mentioned within the context of this thesis, those cited in other scholarly works published by the author of this thesis during the course of this study, and those identified through other critical reviews conducted as part of the authors research efforts, such as those found in the SLRs; [211] [176] [282] [158] [34] [102].

As outlined in the introductory section, the subsequent analysis of the SF models is structured into two distinct segments. Initially, these models are categorized and scrutinized through a multifaceted lens, assessing them within various established categories pertinent to the SF domain. Subsequently, the discussion reconnects with the introductory elements by linking the reviewed models to the proposed adapted NLP strategies employed therein.

The statistics, figures, and diagrams shown here are not representative of all SF research. They reflect only the sample at hand and shouldn't be interpreted as a ranking or preference across the entire field.

This section summarizes features and methods that have worked in prior studies to provide context. The analysis also reviews how alternative models are designed. This enables a comparative evaluation on consistent criteria.

It also catalogs aspects identified as relevant elsewhere in the thesis. This clear identification makes it easier to refer back to these factors in later discussion.

General Approach Following the established notation and definitions of 'fundamental' and 'quantitative' analyses, the ML models under consideration have been classified into four broad categories. In both ML research and economics, it's often difficult to categorize methods consistently using clear, objective criteria. For instance, certain technical indicators such as the RSI ostensibly extend beyond the basic data available in straightforward stock price analyses, e.g. OHCLV data. Despite this, considering that these indicators are grounded in stock price time series (denoted as X), they may yet be classified within a quantitative framework. Conversely, numerous numerical metrics, including currency conversion rates and interest rates, might initially be perceived as external contextual elements denoted as Π. Nevertheless, the need to keep these metrics up to date turns them into elements of a dynamic time series, making them more compatible with quantitative methods.

Furthermore, non-euclidean data, such as that represented in graphical formats, is often categorized as fundamental. However, the practical application of such data outside the scope of quantitative analyses is difficult to envisage.

To avoid getting caught up in a complex debate that's largely irrelevant to this thesis—and more appropriate for economic theorists—the following categories are used:

- Time Series (TS): $\mathcal{P}(.|X)$
- Time Series with Graph / Relation Information (TSG): $\mathcal{P}(.|\Pi, X)$
- Time Series with Texts/Audio (TST): $\mathcal{P}(.|\Pi, X)$
- Texts/Audio (T): $\mathcal{P}(.|\Pi)$

which are intended to enhance the clarity and utility of this thesis.

The categorization of the literature pertaining to the respective approach is delineated in Table 2.1. Additionally, the approaches are illustrated in Figure 2.1. In the literature, several publications such as [168] [113] have reported that T-approaches generally exhibit superior performance. For instance, the studies by [143] [53] [27] advocate for the integration of textual data to navigate the intricate characteristics of stock time series data. An ablation study, detailed in [146], evaluates the predictive capabilities of TST-models by excluding textual data. The findings reveal a significant decrease in accuracy for the TST approach, dropping from 62.69% (ACL-18 🇺🇸) and 53.43% (CMIN 🇺🇸) to 52.88% and 50.69% respectively when solely TS data is employed. Notably, the performance remains relatively more stable at 57.83% and 52.55% when adopting a purely T-approach. Additional instances of ablation studies are documented in [166]. The experiments in [268] reveal that excluding TS data from the model typically results in a marginal performance degradation of only 0.5–0.75%, reducing accuracy to approximately 65%. On occasions, this exclusion may inadvertently enhance performance. Conversely, relying solely on TS data yields robust performance, with accuracy levels fluctuating between 58.9% and 59.9%, consistently trailing approximately 5% behind the optimal results.

In [123], an accuracy of 58.10% is still achieved on the ACL18 🇺🇸 dataset even without the textual data, while on the CMIN-CN 🇨🇳 dataset, a performance of 54.16% is maintained. Conversely, the research presented in [204] posits that the OHLCA[1] prices encapsulate multifaceted dimensions and diverse aspects of the information inherent in stock prices, similarly as argued in [82].

Nevertheless in [82], the authors advocate for TS approaches, arguing that numerous trading patterns encapsulate traders' intentions and behaviors, rendering them applicable across various financial instruments. Furthermore, models designed to capture common market dynamics are expected to incorporate external factors such as T+1 trading[2] or the timing of cash flows, both of which reflect underlying trader strategies.

[1] A = Adjusted Closing Price

[2] Referring to the delay in selling an asset on the same day it was purchased [256].

Table 2.1: Overview for literature by general approach.

T	TST	TSG	TS
[68] [251] [149] [44] [92] [90] [279] [151] [230] [111]	[232] [2] [138] [250] [120] [202] [271] [54] [129] [18] [2] [121] [51] [247] [47] [210] [93] [241] [64] [97] [30] [268] [19] [269] [55] [4] [201] [32] [146] [29] [260] [122] [126] [139] [166] [199] [125] [38] [14] [215] [179] [168] [113] [193] [26] [53] [167] [170] [234]	[238] [96] [240] [255] [231] [249] [229] [244] [108] [24] [200] [66] [169]	[88] [267] [196] [258] [65] [163] [233] [5] [43] [41] [50] [115] [116] [186] [189] [49] [21] [61] [237] [119] [71] [46] [245] [246] [214] [213] [259] [142] [174] [180] [74] [101] [264] [62] [257] [37] [81] [91] [159] [162] [274] [228] [192] [143] [256] [144] [239] [190] [110] [20] [253] [42] [206] [147] [242] [84] [85] [203] [33] [28] [204] [272] [207] [124] [48] [79] [67] [280]

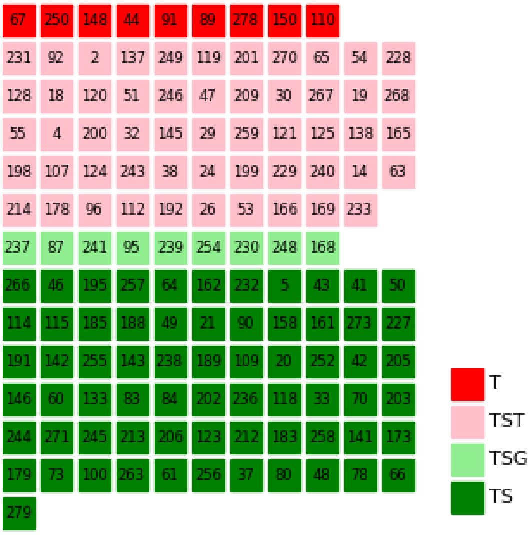


Figure 2.1: Visualization of the different models in their respective category.

In contrast to many researchers, such as [202], who assert that profitability is the primary objective of their models, the stock market is viewed more as an auxiliary problem in this thesis as explained in Chapter 1. Referring to the stronger results of T-approaches serves only to set realistic expectations for performance metrics.

Table 2.2: Overview for publications by task.

SPP / Regression	SMP / Classification	Embeddings	Other
[267] [66] [129] [5] [121] [51] [115] [116] [186] [189] [250] [91] [210] [30] [228] [154] [192] [143] [4] [256] [144] [57] [190] [260] [110] [126] [166] [42] [61] [241] [119] [28] [246] [46] [245] [207] [98] [272] [184] [113] [193] [37] [81] [79] [270]	[258] [229] [232] [251] [129] [18] [163] [233] [149] [44] [43] [41] [92] [51] [138] [131] [247] [21] [159] [90] [279] [162] [274] [268] [88] [238] [19] [269] [55] [256] [239] [151] [32] [260] [146] [29] [20] [122] [126] [139] [253] [199] [108] [125] [200] [24] [122] [61] [93] [85] [84] [203] [237] [33] [25] [168] [111] [204] [14] [64] [215] [259] [179] [174] [74] [101] [264] [257] [177]	[68] [196] [49] [50] [48] [97] [255]	[54] [279] [202] [244] [61] [230] [134] [96] [209] [214] [124] [213] [240] [12] [231] [62]

Selected Task As explained in [175] there are typically regression, movement classification and recommendation (ranking) tasks. Predictive analyses of future price developments are typically performed using tasks such as SMP/SPP. Although risk minimization and portfolio optimization are recognized as mature areas for future development, as delineated in [176], they do not constitute the primary focus of this thesis. The tasks addressed by the model under study are listed in Table 2.2.

In the majority of instances, the forecast horizon, denoted as ω, is set to $\omega = 1$. A predominant line of argumentation, particularly in critiques against generative models, posits that the performance at $\omega = 1$ is sufficiently poor, potentially complicating the prediction accuracy for $\omega > 1$ or $\omega = \{i \in \mathbb{N} | i < \theta\}$. Interestingly, in the study presented in [166], which explores multi-step SMP, it is demonstrated that the model encounters the greatest difficulty in predicting the 3-day trend, whereas it achieves the highest accuracy for the 30-day trend. Similarly, the research detailed in [139] indicates that for $\omega = 30$, trend predictions are significantly more accurate than those for shorter intervals. For volatility prediction, the forecast horizons generally exceed $\omega = 1$ because volatility is by definition contextualized within a temporal context. Additional examples of longer forecast horizons are documented in Table 2.2, as expounded in the accompanying caption.

Markets Numerous studies examine equities across different markets, covering a wide range of countries. Given the huge body of literature in the SF domain, it is important to note that academic studies exist for nearly every national equity market.

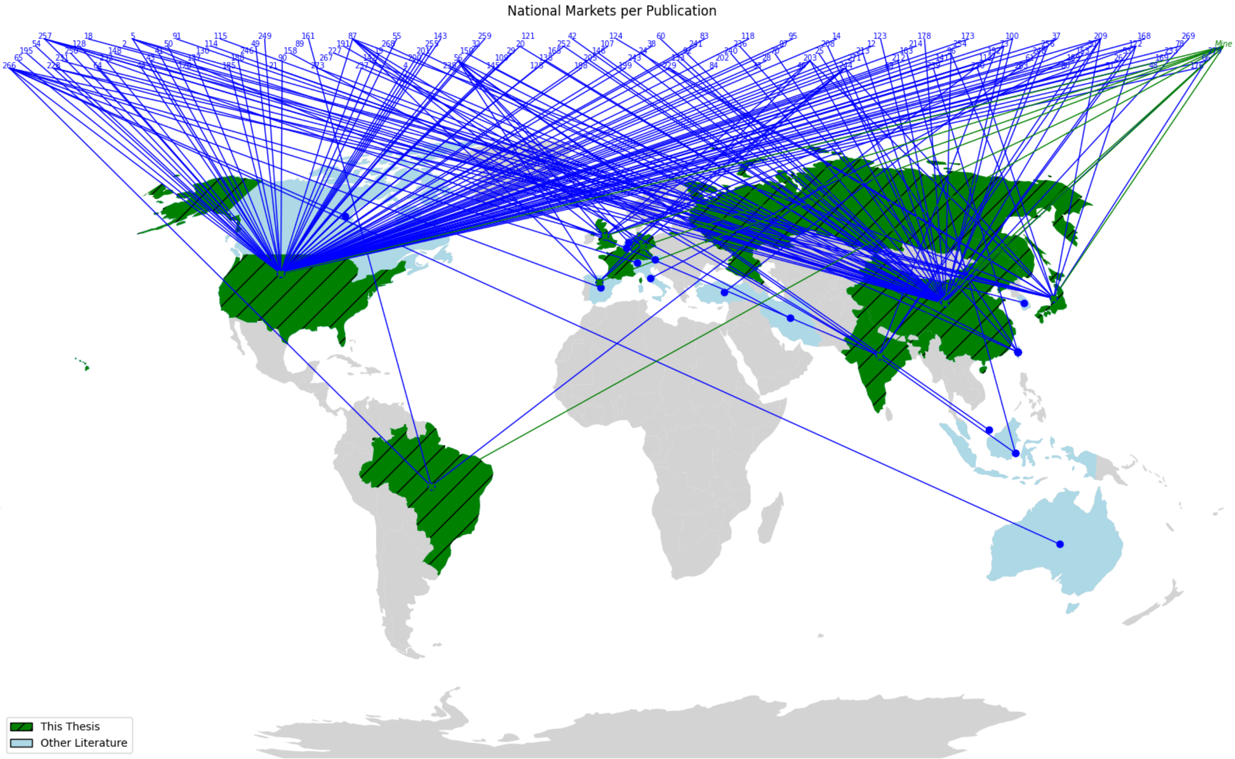


Figure 2.2: Visualization of the national markets investigated in the respective publications cited in this thesis. The indices which represent whole continents were not visualized. This figure was created by the author.

As shown in Figure 2.2, there's a clear focus on certain national markets. Outside the well-covered universe of major U.S.-american stocks, where data is abundant, research tends to concentrate on the home country's national index—likely due to easier access and familiarity. Surprisingly, studies that look at multiple national markets are relatively rare. Even fewer attempt to bring different markets together in a single model to explore cross-market relationships. This highlights a major gap, especially when it comes to building a general-purpose, pretrained foundation model for quantitative stock analysis. Much like LLMs in NLP, such a model should be trained on a broad and diverse range of markets to reflect the variety of market behaviors and interconnections.

Interval Granularity In the domain of financial data representation, stock data is predominantly encoded using OHCL or, less frequently, OHLCV sequences, as detailed in [204]. The granularity of these intervals varies, commonly set at 1, 5,

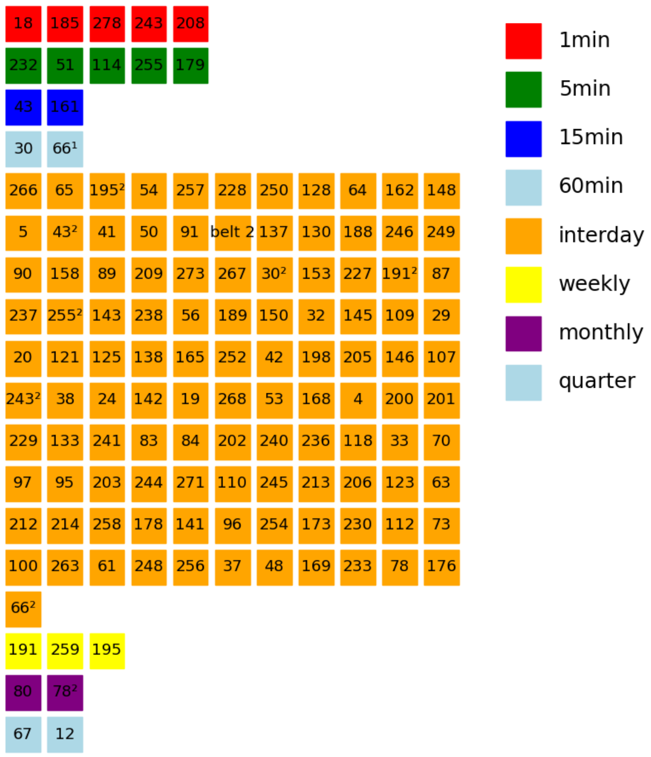


FIGURE 2.3: Visualization of the used interval granularities of the respective publications.

15, 30, 60 minutes, or spanning interday periods. Less frequently used are LOB data, which can represent at much shorter intervals like in [61].

Acquiring intraday data for non-U.S. stocks presents notable challenges. While databases like Yahoo! Finance offer data at 1min intervals, their historical depth pales in comparison to that provided by the AV database, often resulting in insufficient data for training robust models. The frequent use of interday data appears driven by these data constraints.

Furthermore, research referenced in [256] and detailed in [266] asserts that intervals of no less than five minutes are requisite for conducting stable analyses.

In Figure 2.3, an overview illustrates the granularity of intraday intervals utilized in the study.

Data and Datasets In the stock sector, the availability of pre-compiled datasets is markedly less prevalent than in other domains. This scarcity leads to diverse data and processing choices, which in turn produces substantial heterogeneity in comparability. These effects are examined in detail in the following sections.

Dataset Partitioning For the analysis of conventional time series data, diverse strategies exist for dataset organization. A significant number of publications in the SF area adopt a methodology where the initial portion of the time series data is allocated as the training set, followed by an intermediate interval serving as the validation set, and the concluding segment designated as the test set.
This split is widely used, for example (but not limited to) by [43] [92] [199].

Dataset Sources Most studies rely on a small set of homogeneous providers. To keep the overview in Table 2.3 concise, each source below is annotated with a one–line description, and references are trimmed to at most three representative papers per source.
Standardized datasets are notably scarce within the domain of TS analysis, a phenomenon that may be attributed to the readily available nature of stock data which facilitates the creation of customized datasets aligned with specific research interests. Among the more prevalent datasets, the ACL18 🇺🇸 dataset—also recognized as the Stocknet dataset—and the KDD17 dataset 🇺🇸 stand out. These datasets have been employed in various studies, including [268] [199] [29], while the CMIN dataset is referenced in [146] [84] [85].
A more recent iteration of the ACL18 dataset 🇺🇸, covering data from 2020 to 2022, is created and explored in [111]. Additionally, a dataset derived from **NASDAQ** 🇺🇸 and **NYSE** 🇺🇸 sources is developed for use in [66] and is also employed in studies such as [62]. For the analysis of LOB features, the **SSE STAR MARKET** 🇨🇳 dataset is utilized in [61]. Furthermore, the CIKM18 🇺🇸 dataset finds application in [215] [53].
The utilization of standardized datasets, particularly those encompassing T data as seen in ACL18 🇺🇸 and KDD17 🇺🇸, is more frequent for T-approaches. This might be linked to the complexity and challenge inherent in generating large and robust textual datasets, as opposed to the more readily compiled TS data.

Handling Stock Data Several challenges in analyzing stock data have been outlined earlier. These challenges are complex and involve both internal and external factors. Internally, the stochastic behavior and non-stationarity of stock prices present major challenges for analysis. Externally, the frequent absence of values complicates data processing and analysis.

Missing values are common in intraday data, which is surprising given that trading/stock exchanges are mostly electronic. In some instances, these missing values can be attributed to periods of zero trading; however, such an explanation remains questionable for large corporations in the United States or China, where such gaps also occur. The literature rarely discusses this, raising the concern that intraday performance may be inflated when models rely on padding values that are easier to predict.

In the study [256], one of the rare discussions in the literature addressing intervals characterized by zero or no trading—hence exhibiting no or minimal price movement—is observed. This work excludes days where specific time blocks record no trades and acknowledges the potential for misleading SMP accuracy due to prices with little or no movement. Furthermore, the research in [125] considers price movements significant only if they surpass a predefined threshold in the hourly standard deviation. Similarly, the analysis in [19] recognizes minor price fluctuations as a contributing factor to poor intraday trading performance.

An additional factor contributing to the prevalence of incomplete data arises from the temporal dynamics of stock existence, notably IPOs and deslistings. Stocks enter and leave the sample over time, so padding is often applied outside their active periods.

A notable exception is [204], which states that each stock is fully represented; many other studies include incomplete series or omit this detail. For instance, [12] highlights the occurrence of missing data within the S&P-500 🇺🇸 index, particularly prior to 2010. In this context, linear interpolation was found to work best in the authors experiments and is adopted in this thesis. In [91], missing values are addressed by substituting them with the preceding day's data, whereas [237] opts for zero padding. In contrast, [213] filters out three stocks from the ACL18 🇺🇸 dataset due to data omissions in 2016. A similar selection criterion is employed in [20], where stocks that were publicly listed after 2010 are excluded from analysis, with missing data points being padded using the previous day's values. Moreover,

[240] reports the exclusion of certain stocks from the Taiwan stock market dataset, attributed to substantial data deficiencies. [142] adopts a methodology closely aligned with the one used in this thesis, focusing on S&P-500 🇺🇸 companies and retaining only those with complete data for the year 2010. Finally, [119] implements a selective approach towards stock inclusion to mitigate the impact of missing values on the analysis.

Addressing stochastic data represents a formidable challenge in SF, particularly for TS methodologies. The existing literature proposes several strategies to manage these complexities, which will be examined in detail herein. One effective approach to mitigate stochasticity involves the utilization of synthetic data. For instance, as highlighted in [84], training models on such data can equip them with a form of 'meta-knowledge'. This parallels the idea of extending time series lengths to help models develop a deeper understanding of shifting patterns, improving their ability to handle and predict stochastic changes. A similar approach is adopted in [110]. Both [241] and [84] contend that relying solely on historical data may induce overfitting and fail to embed an understanding of stochasticity within the model. Consequently, [84] advocates for the incorporation of artificially generated noisy data samples, supplementing the original dataset. This intervention is applied to the latent representations rather than directly to the initial data, as empirical evidence suggests that introducing noise at the data level does not facilitate model training.

Diverse methodologies have been explored for generating data within the context of missing information, employing models that are distinctly motivated by the absence of complete datasets or size of the datasets. Notable among these are the techniques outlined in [237] [71] [85], which utilize models explicitly designed to address data gaps rather than stochastic variations. In contrast, the study presented in [33] advocates for the preferential use of probabilistic models such as those described in the SDM (see Section 9.1) approaches or in [157]. This recommendation is based on their ability to effectively manage the inherent stochasticity.

A significant challenge in ensuring model stability pertains to the non-stationarity of stocks. A widely recognized strategy to mitigate this issue involves the utilization of returns, or more better, relative returns, as suggested in [275]. This approach is predicated on the expectation that such measures will exhibit consistent characteristics over time. Furthermore, [96] and [125] say that price fluctuations within stock time series behave stationary. However, empirical research within this thesis has identified the use of RLR as one of the most effective methods in this context.

In [119], the utilization of log returns is advocated for time series comparability due to their capacity to encapsulate the compounding effects characteristic of return growth, commonly modeled as geometric Brownian motion within the domain of quantitative finance. Notably, several models incorporate log returns, including but not limited to, the models presented in [119] [245] [272](in conjunction with GARCH models, as is customary in practice) and [162] [30] [98] [245] [98]. Contrastingly, alternative methodologies such as those detailed in [88] employ max-min normalization, leveraging statistical measures like standard deviation and mean for regularization purposes.

Time Periods The temporal spans delineated by the referenced studies are illustrated in Figure 2.4. Access to the AV database enables comprehensive coverage of all intervals from the year 2000 through 2023.

Lookback Window Size The quantitative prediction of future stock prices incorporates historical price data, with the length denoted by Δt, which varies across different models. Figure 2.5 illustrates the range of Δt values employed by various models. When determining the appropriate Δt , two primary lines of argumentation are considered. Utilizing longer Δt values may enable models to better capture and comprehend broader contextual information and identify coarse-grained dynamics or trends. Conversely, if Δt is too long, it may blur important signals, making short-term forecasting easier—but not necessarily more meaningful—which can affect profitability in different ways depending on the context. For instance, the study by [174] contends that for accurate predictions of the subsequent day's price movements, the lookback window should not be overly

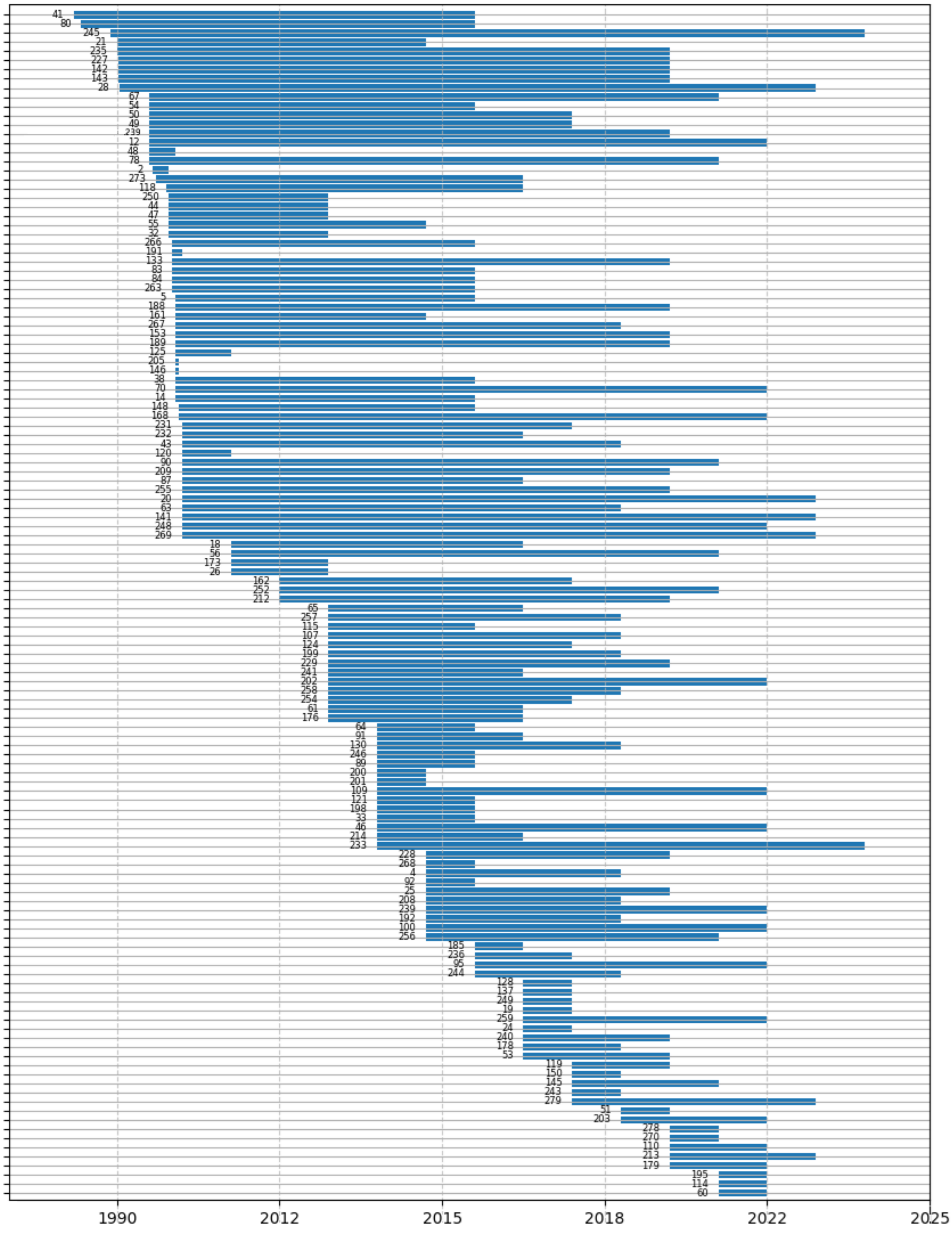


Figure 2.4: Illustration of the time periods shown in the datasets of the publication. To simplify matters, the earliest start year used in the publication is taken as the start and the last point in time as the end, even if there may have been interruptions due to the use of multiple datasets.

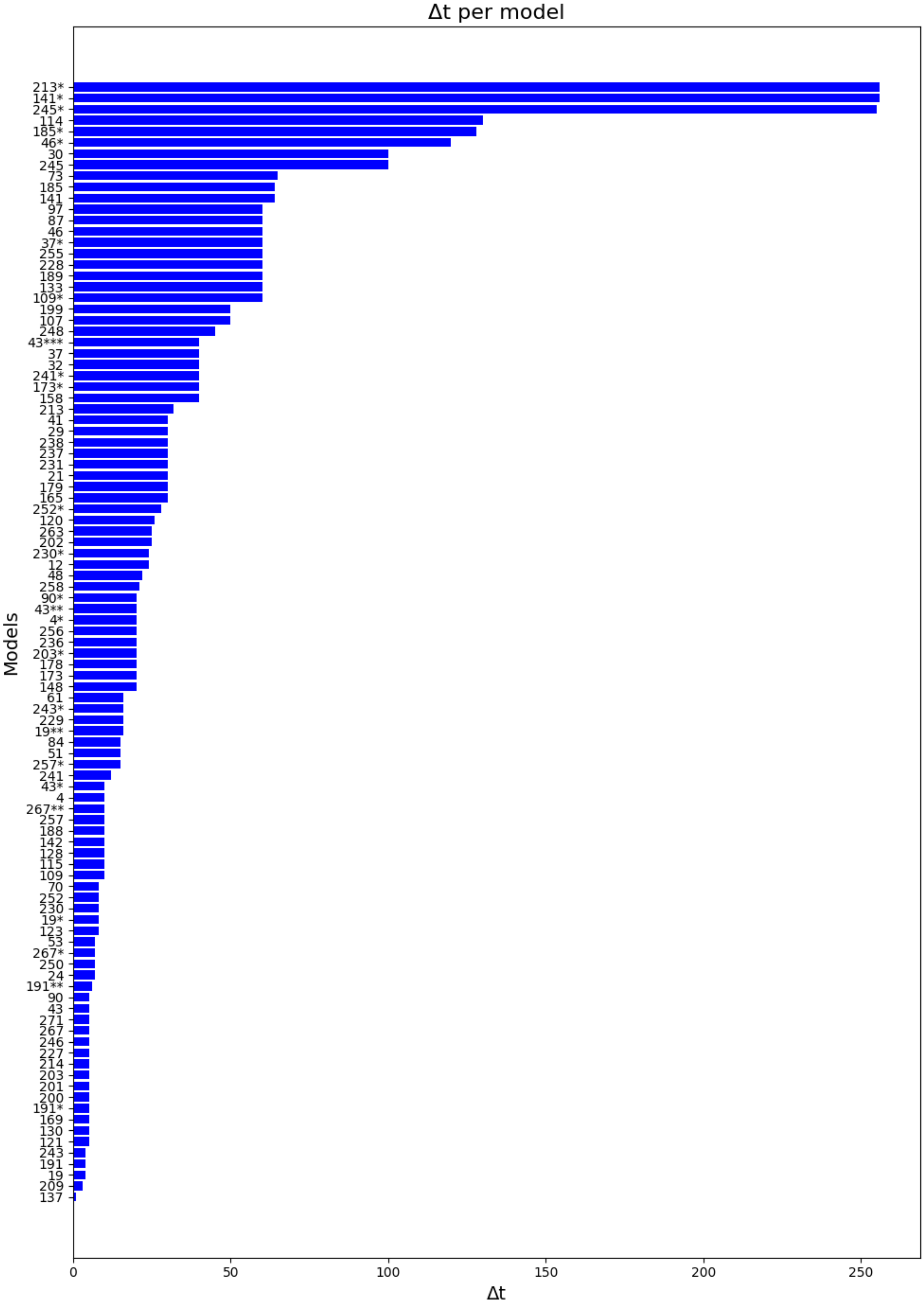


FIGURE 2.5: Visualization of different Δt values from the publications. Note that the underlying granularities vary between interday and intraday, meaning the same Δt can cover very different total periods.

extensive, as this may impair the accuracy of predictions. Similarly, the findings of [62] suggest that an optimal length of the lookback window is crucial; a duration that is too brief may lack adequate information, whereas an excessively long window can escalate computational costs and diminish the model's ability to detect early, informative patterns. In their investigation, [203] underscore the importance of employing multiple time windows to enhance the accuracy of stock movement predictions, noting that stocks exhibit momentum across varied time scales. [96] conduct an analysis to determine the optimal length of the lookback window for their model, concluding that a 20-day period is most effective, a finding that is consistent with established strategies in [1]. Similarly, [257] also identify 20 as an optimal window length, though they experiment with various durations. Furthermore, [20] highlight the need for categorizing models based on their suitability for long-term versus short-term forecasting, indicating distinct methodologies for each forecasting horizon. The authors assert that long-term forecasting primarily concentrates on macroeconomic trends, whereas short-term forecasting is designed to respond swiftly and flexibly to unforeseen events that may cause only minor price fluctuations. In [19] it is demonstrated that the size of Δt significantly influences model performance.

Evaluation Methods The evaluation of the models can be divided into two areas, at least for the SPP/SMP models. The simple performance measurement metrics such as accuracy [24] [38] [244] [38] [125] [147] [29] [20] [166] [122] [253] [127] [146] [61] [85] [84] [33] [25] , F1-Score [38] [244] [200] [199] [226] [29] [20] [166] [139] [127] [25] , MCC [199] [29] [122] [253] [139] [146], [127] [85] [84] [33] , MAE [115] [91] [228] [192] [143] [4] [144] [61] or MSE / RMSE / R^2 [286] [110] [260] [166] [61] [241] [119].

In [274], the reported accuracy values are presented in relation to the defined trend constructs rather than directly to price movements. For the various Chinese indices, accuracy rates between 81% and 83% are achieved. Similarly, an accuracy of 80% is observed for the S&P-500 🇺🇸, while the DJI 🇺🇸 demonstrates an accuracy of 83%.

Furthermore there are stock domain specific measurements methods including the Sharp Ratio [199] [110] [69] [139] [127] [230] [242] [33] [237] the CR [237] the Maximum Mark Down [69] [127] the Return metrics [69] [20], or the IRR [66] [201] [230] [242] [33] explained in detail in Section 4.4. In [61] the top-k selection hit rate is used. Some publication such as [68] [229] evaluate their respective model also on risk sensitive metrics [3]. Stock-specific metrics are often based on the potential financial gains that could have been realized if the model's predictions had been used to inform investment decisions.

Simulation Profitability metrics can be derived either through simulations or by directly incorporating profitability into the loss function of predictive models. For instance, [203] argues that incorporating profitability into the loss function is more effective than traditional approaches such as cross-entropy or hinge loss, as the primary objective in stock movement prediction is to maximize trading gains, particularly from significant price shifts.
The primary purpose of these simulations is to assess the model's potential profitability in real-world scenarios when combined with trading strategies of varying complexities. Several studies have employed simulation-based evaluations to gauge the effectiveness of stock prediction models, including [66] [79] [54] [258] [233] [2] [5] [44] [92] [138] [116] [162] [151] [20] [126] [139] [108] [244] [242] [237] with some implicitly or explicitly predicting trading actions.

Time Series Input Features As previously mentioned, stock price data across different granularities is frequently represented using the OHLCV format. In addition to OHLCV, a variety of technical indicators, often derived from these features, are commonly employed in the analysis of financial time series. Given the extensive number of potential features and corresponding research publications, it would be neither feasible nor constructive to exhaustively list them here. Instead, the interested reader is directed to the comprehensive resources provided by AV (`https://www.alphavantage.co/documentation/`) and Financial

[3] Again these are examples and not a full list of all publications using this metrics.

Modeling Labs (https://www.fmlabs.com/reference/default.html), which offer detailed overviews of widely utilized features and indicators.

Alternative Stock Related Tasks Next to portfolio optimization and risk minimization, other relevant predictive tasks are found in the SPP/SMP literature. Ranking tasks in finance involve arranging stocks with respect to their anticipated returns based on predictive models. One notable implementation of such a model is described in [66], which employs a specialized ranking algorithm. Additionally, [201] integrates a ranking network with a ranking loss function to enhance prediction accuracy. Further illustrations of ranking tasks can be found across several studies: [231] and [61] explore 'Top-K selection'; [213] discusses strategies labeled as 'Top-K, buy, sell' and [230] [134] [96] [209] [214] [124] [240] [242] [257] [197] provide similar methodologies.

Recent literature mainly focuses on predicting values for the next immediate time step. This focus is understandable given the significant challenge posed by predicting equities even over short intervals, which diminishes expectations for successful multi-step forecasting, consequently leading to infrequent exploration of alternative methodologies. Nevertheless, exceptions to this trend are documented in (but not limited to) several studies, such as [246] [134] [214] [249] [113]. Notably, [110] addresses the rationale for multi-day predictions by noting the regulatory requirements imposed on institutional investors. Specifically, it notes that financial regulators require institutional investors to maintain a liquidity horizon of at least ten days for selling risky assets,.

Backtesting Simulation Strategies in Stock Prediction Models Backtesting (simulation) is a crucial step to evaluate how a stock prediction model would perform in real trading. In these simulations, researchers define a trading strategy that uses the model's forecasts to make buy/sell decisions, then apply it on historical data to compute profits. The primary purpose of these simulations is to assess the model's potential profitability in real-world scenarios when combined with trading strategies of varying complexities. Several studies have employed simulation-based evaluations to gauge the effectiveness of stock prediction models, including [66] [79] [54] [258] [233] [2] [5] [44] [92] [138] [116] [162] [151] [20] [126] [139] [108] [244] [242] [237] with some implicitly or explicitly predicting trading actions. Across the surveyed works, backtesting is implemented as a (daily), rules-based portfolio simulation that maps model outputs to trades with fixed rebalancing. The dominant protocol is a closing price-to-closing price Top-k strategy: at each day t, stocks are ranked by predicted return or rise probability, the Top-k are equally weighted, bought at the closing price of t, and sold at the closing price of $t{+}1$, with daily rebalancing [66] [108] [244] [242] [139] [20] [79]. Some studies explicitly compare different values of k to analyze the impact of portfolio size on returns [66]. Classification-based models translate probabilities into trades by ranking confidence scores or by thresholding to execute only high-confidence signals, typically evaluated with the same daily horizon and rebalancing [151] [233] [258] [116] [138] [2] [44] [5]. A subset adopts an opening price-to-closing price intraday round-trip with full reallocation each day [92]. Action-centric methods let the model output *buy*/*hold*/*sell* decisions directly and backtest by executing those actions with variable holding periods [237]. Most simulations are long-only, assume sufficient liquidity, and either use a fixed per-day budget (no compounding) or reinvest the full portfolio each day (compounding); transaction costs are often ignored or modeled as a constant per-trade rate [66] [92]. An alternative binary trading rule predicts $\hat{y} \in \{0, 1\}$ for each stock and goes long on $\hat{y}{=}1$ while holding cash or shorting on $\hat{y}{=}0$, with long-only implementations being most common [162]. News- or factor-driven studies couple text-based signals with prices but evaluate under the same Top-k or thresholded trading rules [126] [54]. Overall, the literature explicitly adopts or implicitly aligns with the Top-k, equal-weight, daily rebalanced, closing price-to-closing price template popularized by [66], with variations chiefly in entry/exit timing (closing price vs. opening price), confidence gating, action granularity, and the treatment of compounding and costs.

Initial capital and position sizing: Simulations usually assume an initial capital (e.g. \$100K or an arbitrary unit of 1.0) and track its growth over the test period. To isolate the model's effectiveness, many studies normalize the investment per trade or per day. As noted above, one simple method is to invest a fixed amount each day or per stock. Feng et al. [66], for instance, reset the investment each day to a constant (e.g. $50K per day) rather than reinvesting profits, so that each day's result contributes equally to overall returns. This avoids compounding and "temporal dependency" during testing, ensuring a fair comparison between days (each day is like an independent trial with the same stake). On the other hand, many researchers do allow compounding by reinvesting gains, which is more reflective of real portfolio growth. In those cases, the portfolio value is updated each day and then fully reallocated according to the strategy. Hu et al. [92], for example, simulate a portfolio that is rebalanced daily: at each morning they allocate the entire current portfolio value evenly into the top-k stocks predicted by their Hybrid Attention Network, then sell at day's end [92]. This means profits (or losses) from previous days affect how much is invested subsequently. Both approaches – fixed daily budget vs. reinvested portfolio – are used in the literature, with the choice often depending on the metric being used (some metrics sum daily returns assuming fixed investment, while others compute actual compounded growth).

Transaction costs and practical constraints: To make backtests more realistic, some studies incorporate transaction costs like brokerage fees or slippage. For instance, the news-driven model of Hu et al. adds a 0.3% transaction cost for each trade in their simulation [92]. This cost is deducted when buying and selling, reflecting broker fees or bid-ask spreads, and it can notably reduce net returns if the strategy trades frequently. In their results, using too small a K (e.g. top-20 stocks) led to frequent trading and the accumulated costs offset some gains. By contrast, Feng et al. [66] ignore transaction fees in their backtest, reasoning that modern U.S. broker fees are very low (around $5 per trade). Similarly, most academic studies assume sufficient market liquidity – i.e. the strategy's trades (often using end-of-day prices) can be executed without moving the market price. Short-selling, margin, and other advanced trading aspects are usually not included unless the study's focus is specifically on those mechanisms. In summary, the simulations tend to be idealized – they trade at published prices (open or close) with either no or minimal fees, aiming to isolate the model's predictive power in a frictionless setting.

Evaluation metrics: After running the simulation on the test set, researchers typically compute a set of profitability metrics. The most direct measure is total return, i.e. the percentage increase in portfolio value over the test period. This is often reported as cumulative return ratio or IRR, calculated by aggregating daily returns across the evaluation horizon. Another widely used metric is annualized return, which normalizes the overall profit to a yearly rate, making results comparable across test periods of different lengths. In addition, many studies benchmark their strategies against standard baselines such as major market indices (e.g. S&P 500 🇺🇸, Dow Jones 🇺🇸) or an equal-weighted market portfolio. Such comparisons reveal whether the proposed model provides added value beyond simply following the general market trend. Some works also include an oracle strategy, which selects the best-performing stocks in hindsight, to provide an upper bound on achievable returns and contextualize model performance.

Aside from return percentages, a few works evaluate risk-adjusted metrics. Although not always reported in the cited papers, it's common in finance to consider the Sharpe ratio (return vs. volatility) or maximum drawdown (largest peak-to-trough loss) of the strategy. These help assess if a high return comes with unacceptable risk. For instance, a strategy that doubled the money could still be less attractive if it had huge swings or large interim losses. In the given references, most emphasize raw returns and sometimes volatility implicitly through observing the equity curve. Hu et al. present an equity curve (cumulative profit curve) for the portfolio over time, which allows visual inspection of volatility and consistency of gains. Generally, a smoother upward curve is preferred to a wildly fluctuating one. Some papers also report the number of winning trades vs losing trades, or average return per trade, but these are secondary to overall ROI in most ML-oriented stock prediction papers.

Using model predictions in simulation: The way model outputs are turned into trades can differ slightly by paper. If the model outputs a predicted return value (regression), one naturally ranks stocks by this value (higher predicted return is seen more attractive). If the model outputs a probability of rise vs. fall (classification), some studies use the probability score directly for ranking or as a confidence threshold. Other researchers might set a threshold on the probability – e.g. only trade if the model is more than 60% confident in an upward move, otherwise stay out. This can reduce false signals and was employed by some to improve precision (especially in cost-sensitive approaches like [151], where avoiding bad trades is emphasized). Indeed, Man et al. [151] introduced a cost-sensitive ensemble of BERT models for news-driven trading; their strategy selects trades where the ensemble has strong agreement, yielding a higher return on investment (reported around 21% in their experiments) at the cost of fewer trades. In essence, these techniques adjust the trading frequency: a lower threshold (or always trading the top prediction) maximizes usage of predictions but can include noise, whereas a higher bar for confidence yields fewer but potentially more profitable trades.

It should be noted that some research goes a step further and optimizes for profit during training. For example, Zhou et al. [203] argue that using a profit-related loss function (instead of traditional classification loss) can directly train the model to favor profitable predictions. In such cases, the evaluation still involves a backtest simulation to verify actual trading performance, but the model has been explicitly tuned to maximize those trading metrics. This approach extends the standard simulation: instead of only assessing a model's financial returns after the fact, the model is encouraged to generate outputs that perform well within the simulator. Still, even these profit-driven models (and others like cost-sensitive models [151]) rely on the final backtest as the ultimate proof of performance.

Across the literature, backtesting frameworks share a similar technical outline: train the model on past data, use its predictions to trade on a forward test set under a predefined strategy (often daily rebalancing with top-K or threshold rules), and measure outcomes like cumulative return, annualized return, and comparison to benchmarks. The main differences concern the chosen strategy (top-K ranking, classification signals, or direct action outputs) and whether transaction costs are taken into account. Nonetheless, the goal is uniform – to assess the real-world profitability of the model. By reporting these simulation results, studies such as

[66] [92] [151] [237] [233] [258] [116] [138] [54] demonstrate how predictive modeling translates into investment gains. This provides a practical evaluation on top of conventional metrics like accuracy: a model that performs well in backtesting is one that would hypothetically earn money if its predictions were used for trading. Each incremental improvement in prediction (be it through novel network architectures, data sources like news [2] [44] [126], or relational learning [108] [139] [242]) is validated by a higher return or a more robust profit curve in simulation. These consistent backtesting protocols across studies build confidence that the proposed ML models are not just predicting stock movement in theory, but can indeed generate profitable trading strategies under realistic conditions.

XAI Given the substantial financial risks involved, the requirement for explainability of specific prognostications in the SF domain is justifiable. In contrast to SF, XAI has been most developed in NLP. In SF, XAI is still early, with relatively few dedicated studies. Explainability is also required for risk management when model outputs are deployed, as noted in [126].

A central question is how the behavior of complex models can be made understandable to humans. The study detailed in [260] employs the generative GPT-4 [173] model to concurrently generate explanations in natural language alongside the predictions. Similarly, the research in [111] utilizes LLMs for SMP, generating explanations for specific predictions. Comparable approaches to generating explanations are also explored in [215] [80] for reinforcement learning.

In works such as those detailed in [38] [124], visual explanations are employed to elucidate the operational dynamics of models. Similarly, [122] and [251] explore the explainability aspects of models. [55] argues that visualizing attention between stocks (as in the ASMs in Section 7.7) improves interpretability. Furthermore, the depiction of temporal and feature-specific attention maps in [19] marks a significant advancement toward XAI in the SF sector. This can be attributed to the ability to express the relevance of specific features to the output at at each timestep.

Financial Instruments This thesis focuses on stocks, but the literature often studies other asset classes and targets alongside them. For instance, the incorporation of ETFs and stock index data is a common practice, as evidenced in numerous studies such as those by [272] [184] [157] [214] [97] (primarily for evaluative purposes).

As [204] shows, overall market data are integrated with stock-specific data because market moves strongly affect stock interactions. The study posits that 'it is widely known that the overall market movements significantly influence the interactions between the stocks' [204]. Further reinforcing this perspective, [245] emphasizes the importance of analyzing financial markets from a macroeconomic standpoint. Similarly, [124] incorporates global market information as a gating mechanism within the model pipeline, suggesting that broader market data can enhance the predictive accuracy of stock-specific models. [240] leverages links among major institutional shareholders to provide global context for local stock interactions.

Moreover, currency exchange rates (e.g. USD 🇺🇸/EUR 🇪🇺) are another popular alternative asset class that is frequently studied alongside indices[4] and ETFs, as evidenced in the literature by [245] [168] [272] [214] [67] (only Exchange rates). That latter study includes gold and index data and also considers cryptocurrency prediction, which has gained traction recently. Notably, research in [100] [46] [67] [180] underscores the increasing importance of cryptocurrencies in financial predictions. Among these, Bitcoin ₿, Ethereum Ξ, Ripple ⊗, and specific trading platform-related coins such as the Binance Coin (BNB) ◈ are frequently analyzed due to their popularity and market impact, as detailed in [180].

Further extending the scope of financial instruments studied, [238] explores the role of derivatives (financial contracts whose value is derived from the price of an underlying asset, such as stocks, bonds, or commodities) —specifically options and futures—in predictive financial models. In [236] future contracts are used.

[4] National stock indices are not technically an asset class, as one cannot invest directly in them. Instead, one would invest in an ETF or index fund that aims to track their performance. For simplicity, the term will be used interchangeably in the following.

TABLE 2.3: Overview of data sources.

Yahoo! Finance	Retail feed for daily/intraday OHLCV, splits/dividends, and basic metadata; widely accessed via community wrappers.	https://finance.yahoo.com	[64] [68] [260]
Kibot	Subscription service for historical intraday and daily price series (U.S. equities/ETFs).	https://kibot.com	[51] [256]
WIND (Wind Information)	Institutional terminal/database covering Chinese markets (prices, fundamentals, macro series).	https://www.wind.com.cn	[5] [269] [228]
Google Finance	Legacy quote feed, mainly used for end-of-day equity data in academic prototypes.	https://www.google.com/finance	[231]
Quandl (Nasdaq Data Link)	Aggregator for financial and economic datasets with unified download APIs.	https://data.nasdaq.com	[19]
Tushare	API for China A-share markets providing quotes, factors, and fundamentals.	https://tushare.pro	[24] [270]
Baostock	Free Chinese stock data API for historical prices and indicators.	http://www.baostock.com	[134]
EastMoney	Retail portal frequently scraped for Chinese market quotes and news signals.	https://www.eastmoney.com	[241] [179]
Kaggle	Community-hosted CSV datasets; often used for benchmarking equity/crypto time series.	https://www.kaggle.com	[119] [255]
Binance	Cryptocurrency exchange exposing public, high-frequency trade/quote data.	https://data.binance.vision	[180]
Taiwan Economic Journal (TEJ)	Institutional database for Taiwan: equities, fundamentals, and events.	https://www.tej.com.tw	[25] [240]
Reuters	Professional newswire used for event- and sentiment-based features aligned to price time series.	https://www.reuters.com	[64] [26]
Bloomberg	Terminal and data feeds offering professional market, fundamental, and news data.	https://www.bloomberg.com	[26]
WRDS	Platform providing access to multiple financial databases through a unified interface.	https://wrds-www.wharton.upenn.edu	[214] [26]
CRSP	Canonical U.S. equity returns and events database used for robust backtesting.	https://www.crsp.org	[81]
Alpha Vantage (AV)	Commercial API for daily/intraday equities and FX	https://www.alphavantage.co	[260]
Company relations	Knowledge-graph style sources of corporate entities/links (e.g., Diffbot, PitchBook, Crunchbase, CB Insights, Tracxn, TianYanCha, S&P Capital IQ) to derive stock relations rather than prices.	(various)	[13]

Chapter 3

Critical Analysis and Research Gaps

Existing research on SF is scrutinized with respect to the implementation of NLP techniques. This examination is conducted to substantiate the originality of the thesis and to affirm that the integration of NLP within this context has not been previously explored in such a manner. Furthermore, the connections between the proposed models and existing frameworks are endeavored to be elucidated. This comparison aims to establish that the proposed models from this thesis constitute a substantive enhancement of the current research landscape, offering innovative extensions to established methodologies.

3.0.1 Embedding Space

In [110], it is explained that stock prices, being continuous and changing at high frequencies, are represented as discrete samples drawn from the underlying continuous distribution, which may not fully capture the intrinsic behavior of stock movements. These samples are referred to as 'market snapshots' [220], and instead of a continuous time series, a concatenation of discrete samples is inserted as explained.

Further in [110], it is claimed that embedding the data into a lower dimensional continuous space, enables the model to learn a more expressive, continuous latent representation that can better handle the stochastic nature of stock prices. This latent continuous space is referred to as the embedding space [196]. The authors argue that this helps improve the generalizability of the model, especially in multi-step prediction tasks where directly predicting noisy target price sequences would further complicate the problem.

A conceptual alignment is established between a market snapshot or a feature vector and word-token embeddings as utilized in NLP. This theoretical correspondence was originally proposed in [270], wherein the latent layer was introduced as a direct substitute for word embeddings. In NLP, it is customary for embeddings to be either pretrained or subject to additional training, tailored to the specific requirements of the downstream task.

Further support is provided by [253], where instability in subsequent modules is reported when multi-dimensional features from multi-view data are used without a stabilizing model. Additionally, [246] posits that financial models exhibit improved performance when operating within their respective embedding spaces. The concept of latent high-dimensional representations has been extensively explored in prior research, including but not limited to [115] [4] [146] [258] [124] [214] [12] [263] [242]. More complex architectures have also been investigated in this context, such as GRUs and LSTMs [29], VAEs [122], and MLP-Mixer-based approaches [253]. Furthermore, [190] demonstrates the application of PCA to OHCLV features, which can itself be interpreted as a form of embedding. For this study, the proposed latent embedding model is treated as structurally analogous to the word-embedding matrix used in NLP.

3.0.2 Adapting Word2Vec

As previously delineated in the literature, such as in the works of [179] [13], numerous types of relationships exist between companies. However, stock forecasting methodologies often fail to adequately represent the breadth of these relationships. Furthermore, as described by [245], it is necessary to capture latent interactions and couplings among financial variables that conventional time-series analysis and basic ML models cannot adequately represent.

The studies in [97] [255] have highlighted the limitations and unviablity of traditional categorization methods, which predominantly classify companies by country or industry sectors. Such fixed and predefined categories are often considered inadequate and of limited use for detailed analysis. The rationale for embedding companies in a high-dimensional space and expressing their interrelationships through these positions is to represent complex dependencies more effectively.

From a non-SMP/SPP based perspective, it can also be important to express relationships between stocks as noted in [217], as finding stocks with similarities is an important task in itself (e.g. for bond swapping, portfolio management, risk management). The latter aspect is particularly crucial in the context of the ASMs discussed in Section 6.11, wherein S2V embeddings constitute an integral component.

In this section, models, approaches, and studies that focus exclusively on the construction of embeddings or relationship expressions for stocks—without incorporating them into downstream tasks or model pipelines—are examined. Given that the identification and utilization of inter-stock correlations and relationships have been established as fundamental components of successful stock forecasting, it is a logical extension to represent these relationships through embedding vectors. It is found that this aspect aligns more closely with the direct adaptation of speech models, as discussed in Section 6.11. The majority of the works reviewed in this section are designed for portfolio optimization and adhere to the underlying principle of generating embeddings that encapsulate company-specific characteristics. The motivation is that a context-sensitive embedding of a stock can help in portfolio diversification, since it can be quickly seen which stocks correlate with each other and have dependent price developments. In stock market modeling, integrating information on stock correlations into predictive models is essential, since these relationships strongly influence future stock movements. At this point, two possibilities emerge: relationship information can either be incorporated into the ML model as a static structure derived from fundamental data, or it can be dynamically generated from quantitative data in the form of correlation metrics.

There are wide-ranging debates about which method is preferable and arguments for both sides. Liu et al. [139] who call the methods ‘price-based methods’ [139] and ‘side-information based methods’ [139] argue, as an example of many others, that quantitative methods can not capture ‘macroeconomic, industry relations, company management, and investor perception’ [139]. On the other hand fundamental methods used to create relational information are not flexible to changing rules, or environments which are present in non-stationary stock data.

In [108], performance differences are demonstrated with respect to the employed relationship modeling, such as "Industry-Product or Material Produced" or "Country of Origin-Country," highlighting the importance of appropriately representing relationships.

The Stock Embeddings model, proposed by Dolphin, Smyth, and Ruihai [50], endeavors to predict companies with comparable returns. The underlying premise is that stocks exhibiting similar return values are likely subjected to analogous market fluctuations, which may indicate a level of underlying similarity. The authors also propose an approach in [48] that leverages the computation of similarities between sliding windows of time series data for two stocks. They determine the probability of a stock pair being classified as a positive or negative sample based on the frequency of their co-occurrence among the top-k most similar time series within each sliding window.

In [168], a bag-of-features approach is proposed for handling time series data, and conceptual similarities to the CBOS adaptation are exhibited.

Another model, introduced by Du and Tanaka-Ishii [54], leverages an attention mechanism that utilizes key and value vectors extracted from financial news headlines, in conjunction with query vectors derived from quantitative data for stock forecasting. The generation of these query vectors involves employing stock embeddings tailored to individual companies, with the explicit aim of training and refining these embeddings.

Sarmah et al. [196] have adapted the W2V training algorithms to generate embeddings for companies by employing random walks of specified length extracted from a pruned graph network, treating these paths as ‘sentence-like structures’. In this graph, individual nodes represent companies, and edges denote correlations in return values.

The Asset Embeddings Model, developed by Gabaix et al. [68], assumes that investors in financial markets organize assets in a way comparable to how documents organize words in NLP. Here, investors, including holdings, mutual funds, and ETFs, are analogized to sentences, with the position of a company—determined by variables such as market capitalization or its proportion in an investor's portfolio—mirroring the position of a word within a sentence. Employing a W2V approach, the model tasks itself with identifying companies with similar positions within an investor's portfolio. Furthermore, [68] incorporates the BERT architecture by adapting MLM to predict a masked company using concatenated company embeddings as input.

In [246], embeddings are also trained and subsequently utilized within the model pipeline. Unfortunately, there is no explicit evaluation of these embeddings, and they are not subjected to a standalone investigation. Through contrastive training methods, the incorporation of negative samples, and specialized loss functions, the model ensures that similar embeddings cluster closely within the embedding space, while dissimilar ones are positioned further apart. Deep hashing is used in [275] in order to represent stock correlations. In the model pipeline in [258] sliding windows of each time series of each stock are represented as dense vectors. Similarly in [139] LSTMs are used to create graph nodes.

The model described in [255] utilizes W2V algorithms to analyze market snapshots. The methodology employed remains unclear, as the paper does not provide specific implementation details, nor is the associated code published. Additionally, attempts to establish contact with the authors for further clarification have been unsuccessful.

Additional stock embedding methodologies developed by Dolphin, Smyth, and Ruihai include the application of case-based reasoning [49]. Furthermore, [47] have adopted a multimodal approach that capitalizes on newspaper articles mentioning multiple companies. Similarly, the research described in [145] [230] examines the co-occurrences of company names in news headlines. The latter ones also proposes to use information about the biggest shareholders per stock. The model in [97] uses static metadata like ‘sectors, company descriptions, and the 3-statement financial data’ [97]. In [217] LLMs are used to extract embeddings using textual data from the descriptions of SEC reports doing an industry classification task.

An intriguing methodology worth mentioning is presented in [218], which adopts an almost ‘antithetical’ approach to embeddings. This research aims to generate optimized NLP embeddings for financial documents associated with companies that may lack specifically defined embeddings in the embedding matrix. To predict these absent embeddings, stock data and stock returns are utilized. Furthermore, [13] details the construction of a knowledge graph that incorporates company (description) embeddings along with 15 distinct inter-company relations. In [209], encoding-decoding techniques are employed to generate high-dimensional dense feature representations for each stock by applying GRUs or transformers to subsections of time series data, with the aim of predicting future stock prices.

The primary objective goes beyond achieving intrinsic evaluation of embedding models, understood as representation quality independent of a specific task [196]. The embedding models discussed throughout this chapter are predominantly designed either to facilitate clustering based on specific sectors [49] [50] [196] [174] or to elucidate market correlations through high-dimensional vector representations. Furthermore, evaluation methods such as those in [24] [50] [97] [196] involve finding nearest neighbors for company embeddings and discussing the connections from an economical point of view. The overarching goal is to enhance extrinsic performance as characterized in [196] by utilizing these embeddings in downstream task. Consequently, the focus is placed on developing an application-oriented implementation that effectively translates NLP problems into the SF domain. An approach to establish objective criteria, proposed in [97], involves searching for similar stocks either within the same exchange or across different exchanges, with the ‘actual’ similarity determined with DWT. Moreover, [255] details prediction tasks based on these embeddings. These tasks include predicting industry sectors, estimating ESG rating scores, or determining company size. [13] offers innovative ideas for evaluating embeddings using expert-labeled datasets. The tasks proposed include similarity prediction, competitor retrieval, and similarity ranking.

Research Gap Based on the reviewed research, it is evident that although several initiatives construct embeddings for stocks, often referencing principles from NLP, the proposed mapping remains underexplored and underutilized.
To the best of the authors knowledge, this work is the first to introduce the proposed SMC and SRE, as explained in Section 6.6, as distinct computational tasks for embedding training. The names for these non-predictive tasks are inspired by [54]. To connect these concepts, the W2V model is adapted to a financial context, using quantitative data both as input features and as target variables. Unlike previous approaches that rely on abstract stock properties, the proposed methodology directly models concrete price movements and absolute price levels, thereby circumventing conventional predictive paradigms. The only potential exception to this assertion may be the work presented in [255], which might exhibit similarities to the X-CBOS/X-SG models introduced in Section 6.6. However, a precise comparison remains challenging due to the limited and imprecise description provided.
Furthermore, to the best of the authors knowledge, embeddings are formally integrated for the first time as a downstream component within the proposed predictive modeling pipeline, particularly in ASMs that leverage S2V embeddings. These embeddings, previously analyzed in isolation, are systematically evaluated here for their effectiveness in modeling relational stock dynamics. Additionally, a structured investigation into the temporal and market-specific axes is conducted for the first time using both dominant W2V paradigms: CBOW and SG. Within the ASM framework outlined in Section 6.11, this paradigm is further extended by implicitly training context-sensitive embeddings during the pretraining phase. This proposed mechanism parallels the use of LLM embeddings (e.g. BERT embeddings) in NLP, yet remains an unexplored avenue in financial applications. To the best of the authors knowledge, no prior study has systematically addressed this gap within the SF domain.

3.0.3 Adapting Hierarchical Models

The preceding sections have outlined how a large body of literature suggests that stock markets exhibit fluctuations and trends at different frequencies. In [276], for example, it is argued that ‘credit and monetary policy cycles’ create periodic patterns, indicating that models able to capture and operate across different frequency components are especially well suited to these dynamics.

The usefulness of NLP for this challenge is examined, while it is acknowledged that suitably adapted LLMs provide strong hierarchical processing internally shown by their capability to process (hierarchically structured) language. As a result, hierarchical models that process data at different frequencies have been repeatedly proposed in the literature. Chen et al. [20] criticizes the employment of excessively fine-grained data, positing that such data may be disproportionately influenced by macroscopic, coarsely granular trends, thereby underscoring the significance of hierarchical data processing. Furthermore, as demonstrated in [202], temporal frequencies are crucial for the processing of fundamental data, exemplified through the disparate temporal representations of news and tweets. Additionally, hierarchical modeling strategies are employed to represent sentences within specific contexts as explored in [55].

Given these considerations, it is logical to employ hierarchically structured models for NLP tasks. However, the authors principal motivation extends to the hierarchical processing of quantitative stock data, which is a focal point of the investigation in this thesis. Support for this approach is provided by numerous related publications. It is explained in [267] that stock prices are shaped by both short- and long-term commercial and trading activities, reflecting multiple trading frequencies.

The necessity of considering multiple time windows to accurately predict stock movements is acknowledged in [203]. This approach addresses the diverse temporal momentums observed in stock movements, which have significantly enhanced their predictive results. Furthermore, the study in [43] not only acknowledges the hierarchical structure of time series data but also, through the implementation of hierarchical attention mechanisms, enables a visualization that highlights the relative significance of data derived from varying time intervals. In [280], the identification and integration of distributional shifts are highlighted as essential components of a model designed for the separate analysis of time series across different hierarchically decomposed frequency levels, distinguished into trend and seasonality components.

Wang et al. [230] acknowledge that stock market fluctuations exhibit regularities across various short- and long-term time horizons. To capture these multi-periodic price features, Wang et al. employ a Hyper RNN-Unit. LSTM networks, as modified in [267], include specialized memory cells that decompose incoming data across multiple frequencies. Similarly, the HATR model described in [229] utilizes a hierarchical structure of stacked convolutional layers to discern patterns at different frequencies. Wavelet transforms are utilized in [5] to represent various frequency components and to denoise the data. The study in [4] considers different time scales as distinct modalities. It employs embedding encodings for these time scales, which are described as 'a time-sensitive version of positional encodings used in transformers' [4]. Choi et al. [28] introduces a model based on the FEDformer architecture [277], which is explicitly designed to handle data across different frequencies. This model integrates seasonal-trend decomposition to analyze more finely-grained structures. In [214], the authors utilize a model that separates the seasonal and trend components. They employ a transformer architecture with varied attention mechanisms to leverage intrinsic periodic patterns effectively. Lastly, [231] critiques traditional attention mechanisms for their limited capacity to capture pointwise dependencies and proposes an alternative approach aimed at recognizing broader, higher-frequency patterns critical in financial contexts. This novel approach suggests a significant shift in modeling techniques to better understand the complexities inherent in financial time series data.

Other frequency and hierarchical processing methods are also explored in various works, including those by [115] [276] (Fourier modification in the attention mechanism), [259] [20], as well as through the hierarchical VAEs in [110], series composition in [166], multi-level contexts in [258], and the concepts introduced in [19], namely the multi-temporal pyramid and the time-wise embedding matrix.

In [244], various graph networks operating at different hierarchical levels model distinct trend levels within the financial markets. These networks are structured to represent individual stocks, specific industries, and the broader market. Unlike frequency-focused approaches, the hierarchy in [244] is based on stock classifications, with individual stocks at the base, followed by sectors and then the overall market.

In classical trading, multi-timeframe analysis is widely used, with indicators applied across time scales to assess market conditions. As described by Lien, prominent indicators such as the RSI, Fibonacci retracement levels, and Moving Averages (e.g. the 20-day or 100-day SMA) are integral to this analysis [133]. Consequently, models that incorporate these technical indicators, effectively engage in hierarchical processing, integrating information across multiple time scales to enhance predictive accuracy.

Research Gap While the research of models that process stock data at different frequencies is well represented in the literature, the proposed CWRNN model that is anticipated has not yet been implemented for stock data. What distinguishes Clockwork RNNs from existing hierarchical or frequency-based approaches is their inherent modular structure: hidden units are partitioned into distinct modules operating at different clock rates. This mechanism allows the model to naturally capture both short-term fluctuations and long-term trends without requiring explicit frequency decomposition techniques such as Fourier transforms or wavelet analysis. Despite their proven success in speech recognition tasks, where temporal hierarchies are equally crucial, CWRNNs have not yet been applied to financial time series forecasting. This represents a clear gap in the literature: the potential of CWRNNs to provide an efficient and interpretable way of handling multi-scale temporal dependencies in stock market data remains unexplored. Addressing this gap is a central contribution of this thesis.

3.0.4 Transformer

The most successful NLP models, particularly those categorized as LLMs, are predominantly based on transformer architectures and pretraining techniques. In this section, the work will be contextualized within this framework by addressing related research. An exhaustive review of all transformer-based financial models is not attempted. The transformer has become a standard model; surveying every application would add little to this discussion. Recurrent transformers for processing extended financial sequences in the stock domain are introduced in this work.

To the best of the authors knowledge, there are no existing examples of recurrent transformers being utilized to enhance Δt and facilitate the reprocessing of longer sequences. The only approach identified that integrates LSTMs and transformers is presented in [236]. However, in this case, the architecture operates in the reverse manner: global attention is computed over the output blocks of the hidden states produced by locally operating LSTMs.

Given this novelty, focus will be placed on literature that, on the one hand, supports the potential benefits of using longer sequences and, on the other hand, explores techniques to manage the challenges associated with such sequences. It is argued in [280] that processing long sequences can facilitate handling domain shifts.

Despite criticism regarding the use and effectiveness of transformer architectures for time series analysis, particularly concerns raised about temporal information loss due to the permutation-invariant nature of self-attention, the limited improvements observed with larger input windows, and the increased complexity without clear advantages over simpler models as highlighted by Zeng et al. [261], transformers remain widely utilized in time series and multivariate time series tasks.

A key reason for their continued application is arguably the inherent distance invariance of the transformer model, which can be advantageous in capturing dependencies between market snapshots and stock relationships, depending on the specific implementation. The relevance of temporal attention mechanisms is further supported by ablation studies, such as those conducted by [122] [258]. In contrast, [186] advocate for recurrent architectures, arguing that they offer greater robustness against noisy data characteristics, making them a compelling alternative in certain time series contexts.

Additional support for the use of multi-head attention mechanisms is provided in [134], where it is posited that investors tend to adhere to specific strategies for buying and selling stocks. When a significant proportion of market participants engage in identical strategies, discernible patterns emerge within the stock market data. When multiple patterns are present, a single model may not capture them well. This limitation is said to be mitigated through the use of multi-headed attention.

Furthermore, [165] surveys several transformer adaptations for time series processing. It is stated that unlike words in a sentence, a single timestep has little standalone meaning, making the extraction of local structure important for interconnections. It is also argued that previous methodologies predominantly utilized point-wise input tokens with channel independence, necessitating the integration of stock correlation data as an additional context. Accordingly, robust input representations are treated as a critical step before downstream training in several models.

In studies such as [209] [246] (which also consider datasets divergent from typical stock data such as commodity closing prices or indices related to the US Treasury and inflation) embedding representations are pretrained specific to the task at hand. The technique of contrastive pretraining, as discussed in [257], involves inserting varied sequences into the model to predict their sequential congruence, drawing parallels to the NSP task in NLP. These parallels are acknowledged, and the adaptation and further emphasis of this approach are proposed to facilitate the learning of macroeconomic trends.

Initial concepts for generalized methodologies using transformers for Multivariate Time Series were introduced by Zerveas et al. [263] and further developed by Nie et al. [165], leading to a universal framework for multivariate time series analysis. Key elements of these methodologies include the patching of overlapping univariate time series and the assumption of channel independence, which necessitates additional context for information on stock relationships. Patching is a method that is often followed in time series processing with transformers, for example in [234] as univariate forecasting. These studies also emphasize the importance of initial input denoising [263]. While the datasets used in [165] [263] did not specifically focus on financial data, the core concepts have been adapted for financial applications in other works. Due to the challenges inherent in the representation of inputs, additional solutions are proposed as outlined in Section 6.11. In the portion of this thesis examining the utilization of pretraining and transformers, channel mixing is employed, which necessitates the transformation of input time series or market snapshots into an embedding space that encapsulates stock and indicator interrelationships. One could argue that this approach on its own falls short of fully capturing the complexities of stock relationships. Enhancements to this model will be addressed in the ASMs detailed in Section 6.11.

Inspired by these ideas, the study by Fan et al. [242] employed a transformer-encoder model for stock ranking tasks. Due to the principle of channel independence, each stock feature and individual stock is processed through a separate encoder, with relationships among stocks incorporated as additional contextual information through a graph model. This methodological approach is prevalent in several models discussed in Section 3.0.6.

Fan et al. [61] introduce one of the few generative models employing an encoder-decoder architecture. Prior to stock prediction and ranking tasks, the model undergoes training on a contrastive task, which aims to maximize the similarity of historical representations within the same stocks, ensuring they align more closely than those of different stocks. It has been demonstrated in [61] that a model trained on these dual tasks can achieve comparable performance on specific financial tasks with significantly reduced training data. The denoising objective, which plays a critical role in enhancing model performance, is also employed in [246] [249].

Numerous models leverage transfer learning which involves training models on stock data from a specific market, stock, or index, and subsequently applying the learned parameters to different markets or indices. This strategy is primarily motivated by two objectives in the realm of SF. Firstly, some models aim to abstract general data patterns or market dynamics that could be applicable to other markets. On the other hand this approach seeks to mitigate the lack of sufficient training data and to circumvent the need for retraining models from scratch. In practice these aims are often aligned: models seek patterns that transfer across markets to improve robustness and reduce computation. Hoseinzade et al. [88] employed a transfer learning strategy to distill generalized patterns by training on a primary dataset and subsequently utilizing the acquired weights for a secondary dataset. This approach intentionally avoids overfitting the initial dataset to maintain a level of abstraction. This methodology is predicated on the assumption that financial markets exhibit broadly similar characteristics, a concept also explored in [88]. The underlying motivation for this approach is to capture universal market dynamics during the pretraining phase, leveraging general embeddings for equities and encapsulating overarching trends within the vector space representation. Hoseinzade et al. argue the efficacy of transfer learning, highlighting its advantage in reducing the necessity to retrain models from scratch. In [163], pretraining (inductive transfer) was used mainly because interday datasets are small and because it can shorten training by avoiding training from scratch each time. The methodology involved training on a bigger dataset followed by fine-tuning on a more narrowly defined set of target stock data. However, this approach has its critics, as noted in [85], where the risk of negative transfer is discussed in contexts where the source and target datasets are notably dissimilar. The scarcity of data is a recurrent theme in the literature, with [184] detailing the training of models on the IHSG 🇮🇩 index and their subsequent application to more specialized indices. A similar strategy is described in [242], where a model initially trained on NASDAQ 🇺🇸 data demonstrated enhanced performance on NYSE 🇺🇸 stocks after additional training. The scalability aspect is identified in [150] as the primary motivation for training the model on a diverse set of financial time series to achieve reduced training time in downstream tasks. The pretraining assets include MSFT 💡, AAPL 💡, GOOGL 💡, AMZN 🛒, ETH ♦ / USD 🇺🇸, XRP ⊗ / USD 🇺🇸, LTC Ł / USD 🇺🇸, and ADA ❂ / USD 🇺🇸 , while the downstream task is performed on BTC ₿ / USD 🇺🇸.

Several models address the challenges of data paucity and stochastic, non-static market behaviors through synthetic data integration. The model presented in [71] includes a denoising component to reduce the impact of artificially added noise. Similarly, [65] uses adversarial data to challenge the latent feature representations in later stages of the model. Moreover, the strategy outlined in [249] involves the use of a masking technique within a graph model, creating new training samples for each masking, that delineates stock and industry interrelations. In [128], parts of the graph are masked during pretraining for bond-default prediction. Additional instances of utilizing transfer learning and pretraining techniques are evidenced in [28], where models are initially trained for SPP tasks prior to being fine-tuned for portfolio optimization applications. Likewise, [113] illustrates an approach where SMP is done before fine-tuning on SPP.

As previously outlined, agreement with [34] is expressed, as it is posited that processing longer sequences may be beneficial for addressing the non-stationarity inherent in stock data. Therefore, similar to [34], it is hoped that the time-invariant self-attention mechanism is a suitable approach for this.

In [280], it is argued that transformers exhibit strong performance in handling seasonal irregularities. This supports the anticipation of utilizing transformers in the proposed approach. Furthermore, the study highlights that the handling of long-range dependencies (as addressed by recurrent transformers) has not received sufficient attention.

However, transformer models are generally not suitable for processing lengthy sequences due to their quadratic time and memory complexities.

Despite these limitations, transformers remain advantageous for processing stock price data, primarily due to their self-attention mechanism, as also noted in [242]. To reconcile the need for handling long sequences with the capabilities of transformers, three novel recurrent transformer architectures have been proposed in [224]. These architectures share similarities with the Recurrent Memory transformer model [10], the TransformerXL model [31] and the Block-Recurrent transformers [94], aiming to enhance the handling of long sequences without compromising computational efficiency. In [115], the capabilities of transformer models to model long-term dependencies are emphasized, advocating for the use of recurrent transformer models to address this need. Supporting this approach, publications such as [110] underscore the importance of long-term horizons for effective volatility forecasting, stating that ‘a long-term horizon is crucial for forecasting its volatility' [110].

As depicted in Section 2.3, it is uncommon to encounter larger Δt values; typically, extended time horizons are associated with coarser temporal granularity, such as interday data. In addition to the patching strategy previously introduced, a prevalent method to incorporate longer time horizons at finer granularities involves stacking data frames, as described in [43]. This approach underscores the significance of the Δt values.

The quantitative model presented in [189] integrates a bagging-inspired multi-transformer architecture tailored for volatility forecasting. This model involves random sampling of input data segments, which are subsequently distributed among individual transformer attention heads. The head outputs are then combined. Parallels are exhibited between this methodology and the recurrent transformers introduced, wherein lengthy inputs are segmented into discrete chunks. However, [189] primarily cites resource constraints as an indirect motivation for this strategy, aiming to enhance the stability of the training process through the adoption of a bagging-inspired approach. The stock data model delineated in [207], which employs State Space Models as detailed in [78], represents a relatively recent and promising advancement in the modeling of extended sequences. This supports using long sequences in financial modeling. Despite not being centered on financial data, the study presented in [165] demonstrates that even within the constraints of the transformer architecture, performance benefits accrue from longer sequences. Furthermore, the analysis in [46] advocates for the use of prolonged time windows, identifying optimal performance with 120-day periods. This window is much longer than those in Section 2.3.

Nevertheless, there are scholarly contributions that present counterarguments regarding the utilization of extensive Δt values in relation to noisy data environments. For example, the study in [233] suggests that high Δt values increase cross-entropy measures because of the inherent properties of noisy data. Similarly, [20] supports the preference for short-term forecasting, rationalized by rapidly changing dynamics and the potential for unexpected events. Moreover, the experiments conducted in [268], which employed larger Δt values for interday data intervals such as 5, 7, or 10 days, have demonstrated a decline in performance. It is critical to acknowledge that for all studies arguing against the adoption of large Δt values, these metrics are generally considerably smaller than the values proposed for Δt in the authors research (wrt. some transformer models). In [264], temporal domain shifts and changing market conditions are cited as reasons for caution when incorporating older data, as the rules and patterns learned from past data may no longer be relevant to current conditions.

Research Gap This thesis aims to bridge several critical gaps in the application of transformer models in the context of financial time series processing. To this end, inspiration is drawn from advancements in NLP, and established pretraining methodologies such as MLM and NSP are leveraged. Notably, to the best of the authors knowledge, the proposed adaptation of NSP in the form presented in this thesis has not been systematically explored in this domain. The only comparable approach—though differing in its implementation—is presented in [257]. Furthermore, with respect to MLM, a multidimensional/multi-axis masking framework has been developed, with spatial, temporal, and feature-wise dimensions. This enables a systematic evaluation of transformer architectures under these constraints. Within this model, both classification and regression tasks are systematically analyzed in the context of masking. Additionally, a significant gap persists in the literature regarding the evaluation of transformer-based architectures across a diverse range of downstream tasks. To the best of the authors' knowledge, this is the first approach that processes substantially longer sequences while preserving transformers' time-invariant temporal modeling. This is accomplished by integrating the proposed recurrent transformer architectures, which enable effective modeling of long-range dependencies while preserving temporal coherence. To the best of the authors knowledge, he is also the first to provide pre-trained recurrent transformers for time series in general and quantitative stock data in particular. Moreover, in contrast to many SOTA models, the imposition of channel independence and patching strategies is deliberately avoided in the proposed approach.

These conventional methodologies often disrupt the spatial and relational structure of the input data, thereby obscuring critical interdependencies among financial indicators of different stocks. By maintaining the holistic integrity of spatial and correlation-based information, the representational capacity of transformers in financial time series analysis is enhanced by the proposed model. Finally, an absence of research is observed in which SOTA LLMs are employed as transformer encoders, pretrained on financial data, and subsequently utilized for downstream tasks such as SMP, SPP, or SDM.

3.0.5 Doc2Vec

The conception of the proposed Doc2Vec adaptation involves encapsulating C (see section 6.2) across a temporal span Δt, thereby forging dense, abstracted representations of the prevailing market dynamics. This approach is analogous to the methodology utilized in Doc2Vec models, where documents are encapsulated into high-dimensional embedding vectors. These models are used to improve training for downstream tasks, provide standalone visualizations, and support comparative analysis, similar to Stock2Vec (see Section 6.6). Although embedding market dynamics over extended periods is rarely documented, related strategies that integrate dense vector representations into model pipelines have been reported. Notably, the authors prior work [223] pioneered the concept of generating market embeddings over protracted durations and conducting their evaluations. To the authors knowledge, this remains a unique contribution. No direct predecessors are reported, only related studies use dense vector representations to compress inputs and provide contextual information to models. These methods are typically used to inform the model about broader market trends and to improve interpretation and prediction. Other approaches regulate learning during exceptional situations, which are also targeted in this work via Doc2Vec adaptations.

The utilization of Doc2Vec embeddings is proposed to facilitate learning regularization by identifying exceptional market situations that are significantly distant from normative states and may not generalize effectively. The idea aligns with [92], where difficult examples are first excluded from training and attention is later focused on critical periods. By contrast, exceptional periods are de-emphasized on the premise that the dynamics observed then are unlikely to be reproducible. In related works, such as those in [115] or in [160], a similar methodology to the authors regularization method is utilized, wherein the loss function is 'adjusted according to domain rules to obtain a better network' [115]. In [223], domain knowledge is encapsulated by the assumption that exceptional situations can be identified through the anomalous distances of embeddings relative to other data points. One of the limited instances where dense representations are systematically compared occurs in the study presented in [208], where time series data from cryptocurrencies are encoded, and the reconstruction error is utilized for anomaly detection. The authors work in [223] proposes a similar methodology, employing the distances between vector representations to detect anomalies, thus facilitating the identification of exceptional market conditions. Further, [142] explores the model weights under varying market conditions, which effectively highlighted the extraordinary circumstances during COVID-19 as well as the associated market volatility. In [169] similarity vectors are calculated and dissimilar sections in the data are muted. Modulating weights to reduce the influence of exceptional situations during pattern abstraction—and increasing them once conditions stabilize—closely resembles the learning-regularization method $R(.)$ $(F^{\langle \mathrm{R} \rangle})$ proposed in [223]. The necessity for some form of learning regularization in response to exceptional situations is indirectly evidenced in various studies. For instance, [201] observes substantial performance discrepancies between models applied to US-American and Chinese stock markets. This divergence is attributed to the distinct market conditions during the test periods; specifically, the authors note, 'The China & Hong Kong test period encompasses the 2015-16 China Stock Market Turbulence—a bearish market scenario' [201] whereas the 'S&P-500 🇺🇸 test period reflects standard market conditions' [201]. The implication is not only the acknowledgment of 'standard market conditions' but also the assumption that model performance under these conditions tends to be superior, a hypothesis that was empirically validated in this work. This observation motivates the learning-regulatory approach, which does not necessarily improve performance in atypical markets but is intended to prevent these periods from disproportionately shaping training. The rationale is that the dynamics regulated are likely to be less applicable in future scenarios. In [30], poor performance in terms of SPP is explained by high price fluctuations, which may suggest—albeit to a certain degree—exceptional circumstances.

As mentioned at the outset, the Doc2Vec approach exhibits significant parallels with models designed to generate dense vector representations of segments within time series data. The model described in [267] utilizes internal memory states to encapsulate trading patterns, thereby transforming these into compact, dense representations of numerical data across temporal intervals. Concurrently, the summarization of stock data blocks, particularly concerning media stock prices, is addressed in [51]. Furthermore, the model outlined in [110] employs VAEs to generate 'more complex and low-level latent variables' [110]. Further examples in this field include the decomposition of sequential data into separate components, as shown in [115], and the use of the 'routing-by-agreement' [138] method to classify features. Significant use of AEs for creating dense input representations is documented in several studies, including [46] [81] [247] [208] [214]. In [280], the idea of learning representations for seasonality and trends through an AE is proposed to integrate them into the model. In particular, [214] argues that such feature representations are key to identifying seasonality and trends in the data. For the models in [269], factorization and reconstruction techniques are utilized to generate low-dimensional data suited for subsequent analytical processes. The study in [16] discusses data compression as a method to reduce the computational demands inherent in processing stock data. Moreover, [20] explores models that produce coarser data granularities within the modeling pipeline. Conversely, the challenge of feature sparsity, as highlighted in the literature such as [55], can be mitigated through the application of dense representations like those found in the Doc2Vec adaptions.

The authors proposed Doc2Vec based abstract summaries of market conditions over a period of time can be used as indicators of macroeconomic trends. In [5], AEs are used to learn abstract representations of multivariate data, with macroeconomic information added as contextual input. Similarly, the study in [230] utilizes a graph contrast module to learn macro-market scenarios, positing that this approach could mitigate issues arising from non i.i.d data. In [101], a HMM is adopted to conceptualize the current macroeconomic market conditions as hidden states, aiming to delineate distinct, discrete market conditions through a specialized training regimen dubbed ‘Stock State Modeling’ [101]. The work in [264] introduces a contrastive learning task that leverages embeddings of sequences to discern domain shifts effectively. This methodological framework seeks to train the model on recognizing and adapting to these shifts. The concept of Neighbor Similarity was proposed in [223], based on the premise that embeddings in close proximity are likely to exhibit analogous future values. Although this approach yielded limited success beyond Volume prediction, it provided valuable insights into embedding-based predictive models. Conversely, [264] tackles this issue from an alternative perspective by initially generating embeddings where future price trends serve as labels. The labels, while not directly employed in making predictions, serve to guide the spatial proximity of embeddings within the vector space, with the Frobenius norm facilitating this arrangement.

In the context of fundamental/T data, the concept of generating event embeddings is explored. Such embeddings involve the transformation of specific events derived from fundamental/T data into high-dimensional vector representations. These vector representations effectively encapsulate market dynamics over designated periods or under particular conditions. Event embeddings are considered analogous to the proposed quantitative Doc2Vec adapted embeddings [223]. The literature presents several models that incorporate event embeddings or analogous structures. Notable examples include works by [232] [18] [251] [44] [279] [269] [32]. These models utilize embeddings to interpret and predict market behaviors by encapsulating event-driven market characteristics.

Several models utilizing T data exhibit profound similarities with applications of quantitative Doc2Vec adaptions, particularly in their attempts to contextualize the latent representations in relation to one another. In the model from Ma et al. [149], news events are employed to create event embeddings. Ma et al. underscore the necessity of modeling the relationships between different events to facilitate effective learning, referring to techniques utilized in NLP for Doc2Vec models. Moreover, in [38], event embeddings are integrated into a graph to systematically establish inter-event relationships.

Research Gap Unlike adaptations such as Stock2Vec, the literature reports no dedicated pretraining or evaluation of market-situation embeddings, let alone their integration into ML pipelines. One approach in [121] incorporates a distinct evaluation methodology, yet it remains fundamentally different from the proposed in both structure and assessment. This method employs a hybrid framework that integrates TS and fundamental mixed-method approaches, exemplified by the concept of the ‘investor information space’ introduced. This framework utilizes Tucker decomposition to uncover latent relationships among variables. It is expected that, unlike models that implicitly produce dense vector representations of long time periods, better quality for the intended applications can be achieved by explicitly analyzing the embeddings. As mentioned at the beginning, there are some learning regulation approaches that are similar to the one proposed in the authors prior publication, i.e. [223] but a research gap exists in the regulation of training through the evaluation of market conditions based on their estimated reproducibility/rarity.

3.0.6 Adapted Speech Models

Following [132], foundation models are defined as pretrained, often self-supervised models trained on large datasets to learn general domain representations. These models have become the cornerstone in fields such as natural NLP and CV. The authors suggest that adapting these models to time series data—referred to as TSFMs—offers a promising direction for future research. The argument rests on zero-/few-shot capabilities and broad cross-domain applicability.

Liang et al. highlight examples of general-purpose TSFMs such as TimeGPT [72] and Lag-LLaMA [191], which diverge substantially in both concept and target domain from the proposed ASMs proposed in this thesis. The study categorizes the pretraining tasks for these models into two principal types: generative tasks, which include generative decoder models and predictive next-token prediction tasks exemplified by [220], and contrastive learning tasks, as utilized in NSP/TM [220].

In their perspective paper, Guo and Shum [82] propose the idea of building foundation models for quantitative finance, though they present neither experiments nor a concrete implementation. They term the concept the 'Large Investment Model' [82], by analogy to LLMs. The model is envisaged to be pretrained on large, sector-wide data across assets and markets and then fine-tuned for specific tasks.

Their rationale follows the success of self-supervised pretraining in NLP. However, their focus pivots specifically towards the application of generative models. In contrast, this thesis adopts MLM and NSP as the main pretraining tasks. The utilization of predictive models, akin to BERT-like architectures, is advocated, predicated on the assertion that predictive forecasting alone (in contrast to generative,multi-step forecasting) within the stock domain presents considerable challenges. This complexity limits the practicality of purely generative approaches, as noted in prior work. Rather than generative pretraining on separate univariate series, the approach here incorporates multivariate, relational, and spatio-temporal dependencies among stocks. A pretraining strategy that involves a masking task is employed, as it is believed to better capture the dynamics inherent in financial datasets. For the processing of univariate time series, they propose the adoption of regressive generative tasks and patching techniques for the transformer-based Large Investment Model, methods that are conceptually similar to those outlined in [165].

Because relational data are absent, structured inputs (e.g., graphs) are added during fine-tuning. Consequently, the dimension of relational information is construed as a component pertinent to downstream tasks. Pretraining uses indicators and time-step cues but excludes cross-indicator and cross-series correlations described in [62]. Data diversity is emphasized and is central to the proposed ASMs. This is important because the models are trained on data from various national markets, as explained in Chapter 4. In the study referenced by [67], a foundational model – TimesFM [35] – undergoes pretraining utilizing a variety of financial datasets, analogous to the methodologies employed in [280]. Its generative pretraining targets next-step prediction and is followed by task-specific fine-tuning. The authors report weaker performance without pretraining. The in [82] proposed enhancements for finetuning encompass a range of techniques, notably SMP and SPP, alongside advanced strategies in portfolio optimization and risk management.

For the domain of risk management, a series of advanced methodologies utilizing ASMs is proposed in Section 9.1. Their strength lies in modeling relational dynamics relevant to risk assessment. Additionally, the potential of fundamental investing, also proposed in [82], is explored through the lens of adapted vision and language (V+L) multimodal models. In this setting, ASMs assume the backbone role typically held by LLMs. According to Guo and Shum, the universality of LLMs should be considered in several dimensions when designing Large Investment Models: instrument universality (which is outside the purview of this thesis due to data limitations); exchange universality (which is achievable within the operational constraints); and cross-frequency universality (manifest in both hierarchical models and during the pretraining phase). They ask whether ‘pretraining + fine-tuning’ is feasible for quantitative stock research and suggest that such a paradigm shift could significantly enhance research efficiency within the field.

This thesis presents a constructed and evaluated implementation of a robust investment model. Proposed as an ASM, this Large Investment Model represents the culmination of various explorations into NLP-adapted methodologies. It is envisaged as a progressive stride towards realizing an ‘artificial general intelligence system for quantitative investment’ [82].

In addition to the concept of adapting foundation models for application to time series, particularly within the realm of quantitative stock data, existing methodologies have explored the direct utilization of LLMs for such purposes. Similarly, Wang conducts an empirical study on large language models for asset return prediction, outlining the scope and limitations of such applications [288]. However, these applications exhibit three principal limitations, which this thesis intends to address comprehensively. First, it is critical to acknowledge that instead of utilizing LLMs as foundational models, the prevailing approach employs generative transformer decoders, which, although similar to LLMs, exhibit substantial differences in both function and design. Second, the structural orientation of LLMs typically prioritizes representation learning over advanced temporal processing capabilities, often aligning more closely with embedding strategies from transformer implementations, as outlined in Section 6.11.1. Third, the domain of quantitative finance is conspicuously underrepresented in these models, suggesting a gap in the current methodology and application. Furthermore, the literature, such as [102] [165] [67] [15], frequently highlights the prevalence of channel independence in most transformers used for time series data. This contrasts with the spatio-temporal design of the proposed ASMs.

In the seminal work by Jin et al., as previously referenced in [222], the direct adaptation of LLMs for time series analysis may initially appear counterintuitive. However, Jin et al. highlight that this methodology is embraced in several studies, such as [278] [22] [137], which either employ LLMs or their transformers and self-attention mechanisms. Despite fundamental differences from the proposed approach—such as in [72] (which lacks an LLM backbone) and [191]—mainly due to their focus on generative prediction and pretraining, the use of LLMs is clearly identified as a promising research direction.

Additional applications of LLMs for time series analysis have emerged in non-financial sectors, such as in traffic forecasting, where they are used for generative tasks, exemplified by works like [137]. This approach deviates notably from that of ASMs in several respects: it lacks pretraining, eschews the incorporation of context-sensitive embeddings that are pretrained, and employs frozen components within the model architecture. Moreover, the structural foundation significantly diverges from that of ASMs, as it utilizes a GPT-2 backbone wherein feature vectors, derived from each variable of the multivariate time series, are explicitly provided.

In [15] the motivation for the adapted LLMs (here again GPT-2) (again not for financial data) are the few shot capacities and the self attention mechanism of the LLMs, but the time series recognition abilities are acknowledged as a problem. An example in quantitative finance in the use of LLMs, in this case BERT, for time series data can be found in [136] where the BERT-transformer algorithm is used to predict the market at future states by leveraging its structure to encode transition probabilities, replacing selected values with masked states, and refining predictions through the residual-based transformer framework. The inputs of the model (and BERT) are (regressive) sentimental and illiquidity variables which is therefore remotely similar to the authors embedding-based approaches, in which the LLMs are also treated as special transformer-encoder variatons. The 21 in [136] different sentiment values are not textual data but technical indicators such as EPS.

As previously noted, analyses that solely focus on 'uni-stock movement prediction' [123] without considering intercorrelations among stocks are generally insufficient for accurate stock price prediction. Several studies, such as [240] [123] [200] [249] [197] [179] [66], have criticized the common practice within SF of examining stock trends as isolated time series. The critiques stress the omission of inter-stock relations, which are crucial for understanding market moves.

The investigation of inter-stock correlations as 'multi-stock movement prediction' [123] is widely recognized in the literature as a critical element for accurate SF. Numerous studies underscore the importance of modeling relationships between stocks comprehensively. For example, it has been shown that irrelevant information, such as prices from unrelated stocks, can negatively impact prediction performance [116].

In [53], the limitation of relying solely on historical price data for predicting future trends—rather than incorporating intercorrelations—is explicitly highlighted. This thesis aims to address this research gap by developing models that integrate intercorrelation analysis to enhance predictive accuracy. Through the proposed adaptation of MLM techniques and the contextualized learning of embeddings and relationships, a concept is used that comes close to the idea of 'relation discovery' presented in [123].

The most pivotal argument in this domain is centered around the identification of stock intercorrelations as a fundamental mechanism for predicting stock prices. The concept of momentum spillover is frequently mentioned in literature, e.g. in [230] [26]. This phenomenon describes how the momentum—whether upward or downward trends—of one asset or market can influence and propagate to other assets or markets [3]. Stocks from the same industry or supply chain often exhibit correlated movements, underscoring the interconnected nature of financial markets [240] [174] [230]. However, as stated in [26], not every movement characteristic necessarily has to spill over to other stocks.

Many studies emphasize the value of incorporating stock relationships within predictive models, identifying this approach as one of the most promising [242] [259] [249] [204] [185] [174] [231] [33] [269]. Furthermore, some research initiatives utilize pre-defined relationships based on correlations, industry sectors, or even external databases like Wikipedia to enhance model accuracy [230] [66] [166] [199] [108] [25] (using fund investments to create graph edges). However, this approach has been criticized by several works, which argue that relying solely on predefined relationships is inadequate [242] [174] [124] [259] [275] [26]. It is increasingly acknowledged that the relationships among stocks must be contextualized temporally, recognizing the dynamic nature of these relationships. In the ablation study in [26], different types of relationships, which are also compared in the attention mechanism, are tested and the inferred relations perform significantly better than the predefined ones such as 'supplier', 'customer' or 'competitor' (up to 2.7% SMP accuracy).

Modeling the temporal context is crucial not only for capturing relationships between stocks but also for identifying broader market trends. As noted in [244], different stocks have varying levels of influence on the overall market state, and this influence can change over time. Such modeling is crucial for a comprehensive understanding of market dynamics and for the accurate representation of temporal variations in stock importance.

The main challenge is explained by Chen et al. as ‘There are two major challenges: it is non-trivial to model the relationship between corporations; it is difficult to integrate corporation relationship into existing prediction model’ [24]. This is the first gap/challenge that is intended to be addressed with the ASMs. The authors of [242] highlight the superiority of models that dynamically represent relationships. Consequently, spatio-temporal models have become highly popular for predicting stock prices due to their ability to capture these dynamic interactions, for their efficacy in stock price prediction and the ability to reflect the complex and time-varying nature of stock relationships. The perspectives are concurred with, and it is contended that relationships among stocks are too intricate for static models and must be understood as inherently dynamic. In line with this, the viewpoint is supported that creating an embedding which captures stock price changes across companies over time effectively represents a company’s temporal dynamics [255], and the critical importance of considering temporal contexts in modeling stock relationships is acknowledged.

As posited in Chapter 1 and in [222], LLMs are well-suited for spatio-temporal data analysis. Conceptually, LLMs process data that is sequentially organized, such as word-tokens, which are represented through their embedding vectors positions within a multidimensional vector space [222]. These vectors not only represent the tokens but also encapsulate their interrelationships and, after the addition of the position embedding, their temporal correlations. In the domain of NLP, numerous tasks rely on the sequential processing capabilities of LLMs. These tasks include predicting future developments of an input sequence, such as in next-token prediction, as well as generating comprehensive semantic interpretations from structured inputs, such as sentiment analysis. This requirement is particularly relevant for financial time series data, where stock prices are temporally ordered and exhibit high intercorrelations. Although most LLMs are based on transformer models, which have proven effective in semantic correlation extraction across various SF models, they encounter specific challenges when applied directly to time series data. Criticisms, such as those presented in [261], argue that despite their efficacy in semantic analysis, transformers struggle to capture temporal dynamics effectively due to their permutation-invariant self-attention mechanism. This limitation is a significant concern when transformers are used for SF, as many existing approaches fail to adequately address the dual need to process both indicator and stock correlations within the temporal dimension. These correlations are often treated simply as extra embedding dimensions, leaving it to the attention mechanism to infer inter-temporal relationships—a task it is not inherently designed for. This oversight can undermine the model's ability to leverage the full predictive power of temporal data, thereby impacting the effectiveness of stock price forecasts.

To provide a comprehensive overview and acknowledge research that shares the same motivation, the relevant studies will be categorized into three distinct groups. Firstly, models that exhibit characteristics similar to ASMs will be examined, including the three key capabilities of spatio-temporal processing, the ability to generalize or expand, and proficiency in few-shot or zero-shot learning. The last feature being crucial for managing the dynamic and often unpredictable shifts in data distributions. Secondly, models that employ techniques akin to those intended for use—particularly those that utilize embeddings to represent stocks—will be explored. This category includes models that treat technical indicators as a form of embedding or those that incorporate a global contextual understanding through embeddings. Lastly, models that either utilize LLMs—which diverge significantly from most of the approaches taken in this thesis—or models from other domains, such as vision, that are adapted for SF, will be acknowledged. This segmentation will allow us to delineate the landscape of existing methodologies and highlight the innovative aspects of the authors approach in integrating stock data with LLMs.

Initially, models designed to incorporate attributes analogous to ASMs is explored. The foremost, and perhaps most critical attribute, is the capability to model spatio-temporal relationships. As elucidated in [100], a primary challenge in asset representation is capturing the dynamics at specific timesteps. This issue has been addressed by [124], who developed novel correlation representations for every t (see Section 6.2). Furthermore [230] argue that each stock needs an individual and temporal-dependent representation encoding.

The integration of temporal dimensions and spatial representations of stock relationships has precipitated the development of numerous spatio-temporal models. Predominantly, these models (partially) operate in a non-euclidean space and employ graph-based or process-based graph neural networks to delineate stock relationships [231] [174] [26] [240] [259] [204] [98] [33] [242] [249] [229] [53] [163] (sorting stocks in terms of similarity as an additional context) [139] [25] [108] [244] [123] [79]. Ablation studies in [79] further highlight the effectiveness of graph-based approaches compared to relying solely on time series information.

For these non-Euclidean models, [175] provides a comprehensive conceptual overview that is widely adopted in the GNN-based literature reviewed herein (similar to the overview in [124]). In this paradigm, temporal information is processed through a dedicated temporal encoding mechanism, while spatial dependencies are captured within a relational module, typically constructed using a GNN or a similar graph-based model. In the realm of NLP, parallels can be drawn to CLM, where Π (see Section 6.2) encapsulates relational information i.e. the graph, and $(\tilde{w}^{(1)}, \ldots, \tilde{w}^{(t)})$ represents the time series. The underlying graph structure is typically based on correlation or similarity matrices—such as the Pearson correlation coefficient [175] [79]—or on predefined relational graphs, sometimes built from textual data. As emphasized in several studies, such as [53] [79] [167], it is crucial to account for the temporal aspect of stock data, which can also manifest in dynamically evolving relationships and structural changes within relational graphs.

Alternative approaches which depict a method of working in an Euclidean space include the utilization of tensors [203] / stock correlation matrices [93] [125]. Certain studies, such as [242] [33], assert that employing singular graph structures is insufficient to capture the complexity of financial networks. Consequently, several works have adopted the use of multiple graphs to represent different types of relationships, as demonstrated in [185] [230]. Others have utilized hyperedges / hypergraphs to enhance the expressive power of these models, as for example done in [206] [147]. Specifically, [230] incorporates techniques from the NLP domain—namely, Word Context Factorization and Positive Pointwise Mutual Information—to refine the representation within hypergraphs, thus enriching the model's ability to encapsulate and analyze multifaceted relational data. In [146] Transfer entropy is used to model causal relationships between stocks.

Other models incorporate multiple modules for spatial, temporal, and spatio-temporal dimensions such as [62] [37]. It is widely acknowledged that embedding this relational information within a temporal context is crucial to model efficacy, as supported by [174] [124] [204] [119] [249] [203] [259]. Numerous studies employ DWT to generate time-series dependent correlation information, including [33] [119] (Logistic Weighted DWT) [231] [242] [204] [20] [53] [177].

The capacity to manage distribution shifts represents a critical quality in the adaptation of speech models. Jeon et al. [100] argue that while modern SF models effectively capture common trends, they tend to downplay sudden changes by treating them as anomalies or outliers. This classification approach may result in the oversight of significant market transformations, potentially culminating in substantial investment losses. A key challenge in developing such models is maintaining sensitivity to both long-term trends and sudden market movements. Maintaining this balance is essential to ensure that significant distribution shifts—critical for accurate financial forecasting and risk management—are not overlooked.

As delineated in Chapter 1, it is anticipated that these models will exhibit capabilities akin to few-shot learning, wherein minimal exposure to new scenarios facilitates rapid adaptation. The concept of employing few-shot learning methodologies to address SF is proposed in the study by [264]. This work introduces a mechanism for recognizing domain shifts by analyzing temporal windows, a method that shares similarities with the NSP adaptation in Section 6.9.2 and the Doc2Vec adaptation in Section 6.7. The integration of global context information, as detailed in [264], enhances model predictions by incorporating this broader understanding of market dynamics. The importance of this integration is further highlighted in [245], which identifies it as a key factor in model performance. Studies such as [4] [5] [258] adopt a TST-strategy, inputting global market movements to enable the model to establish pertinent correlations, thereby enriching the predictive framework. In [45], a bifurcated approach is proposed, employing both local models for individual stock assessments and a global model to incorporate wider market data.

As highlighted in [253], the issue of inadequate modeling of stock correlations is addressed through the introduction of an attention mechanism modification, termed the 'Multi Head Market Attention Block' [253]. This modification is designed to enhance the existing model pipeline by integrating a more robust market-stock correlation modeling framework, thereby enriching the analytical capabilities of the model with respect to understanding complex financial interdependencies depended on global market contexts. Similarly, [12] proposes a uniform, multifaceted algorithm aimed at learning invariant representations to effectively address distributional shifts, whether within or across different stocks. Furthermore, [230] outlines additional methodologies to address challenges associated with non-i.i.d. data and distribution shifts. The work discusses the use of causal learning and domain generalization as strategies. The model itself implements hypernetworks and meta-learning to operationalize these strategies. Lastly, [97] explores strategies to recognize and accommodate temporal distribution shifts and non-stationarity in the creation of embeddings used in the model pipeline.

In Chapter 1, the critical need for SF models to be extensible and generalizable to adapt to evolving market conditions has been stressed—a capability that manifests in several aspects. In [263], various domains (excluding stock data) are exemplified wherein transformer-based architectures are employed to develop generalized models capable of managing multivariate time series data. Notably, many SF models suffer from their inability to seamlessly incorporate new stocks, which can diminish their capacity to generalize patterns, particularly in relation to new, expanded, or modified markets as mentioned for example in [88] [79]. The market and the assets in it continuously evolve through events such as bankruptcies or IPOs. The term 'Universal Predictor' [88] used by Hoseinzade et al. aptly describes the envisioned architecture for ML models in relation to SF, a descriptor that fits well with proposed ASMs due to their inherent extensibility.

For LLMs, the theoretical possibility exists to expand the vocabulary by introducing new word-tokens and embeddings [218]. If these embeddings are pretrained, the model can more quickly learn their meanings and better align them with existing tokens. The broad ability of LLMs to handle many different types of text further highlights their versatility. However, the SF models, typically non-expandable and trained for particular markets, lack this desirable feature. Hoseinzade et al. [88] point out that SF models often fail to account for the emergence or disappearance of companies in the market. In contrast, domain generalization is proposed as an effective way to handle distribution shifts and non-i.i.d. data, as discussed in [230]. The potential and utility of such models are reinforced by Hoseinzade et al., who suggest that the 'promising results of this experiments suggest to further investigate the possibility of extracting general patterns that explain the behavior of different markets' [88]. Few studies such as [242] [88] have attempted to train models on specific markets before retraining them on others to assess how the models benefit from transferred knowledge and whether the general patterns learned are applicable across different contexts. Furthermore, the approach of pretraining models on a broad range of stocks before fine-tuning them on a subset is detailed in [163] [150].

As stated in the work of Xu and Cao [245] speech models are adept at 'jointly model high-dimensional dependencies, long-range dependence structures [...] and latent features and relations' [245]. Xu and Cao emphasize the importance of integrating global contextual information (as discussed in the previous section), which is essential for applying models across diverse financial settings.

Regarding similar techniques, following the proposed pretraining approaches outlined in Section 6.9, most attention is given to representing inter-stock relationships through embeddings. Literature in the field, such as [62] [259] [230] [255] [197] [95], underscores the complexity inherent in inter-stock relationships. These relationships, as highlighted, can manifest in multiple forms [179] and their complexity is arguably beyond the scope of the already discussed modeling approaches.

To address these challenges, representing stock relationships within high-dimensional embeddings is recommended. This aligns with Xu and Cao's argument that such embeddings are well suited to capture complex relationships. Notably, [259] identifies two main issues with GNN-based approaches: their limited ability to model the dynamic and asymmetric nature of stock interactions, and the shortcomings of existing stock-to-stock attention mechanisms in capturing temporal correlations. In response to these challenges, the study in [124] proposes representing each stock and time pair as an input vector of dimensionality $\Delta t \cdot |C|$. This approach is echoed in [37], where a flattened version of the time series of dimension $\Delta t \cdot |C|$ is utilized. These transformed representations are then integrated into three distinct modules, each dedicated to capturing either spatial, temporal, or spatio-temporal relations. Moreover, the work in [230] employs contrastive inter-stock training to formulate and refine stock embeddings.

Additionally, [246] incorporates embeddings within its model pipeline, reinforcing the prevalent integration of embeddings in diverse modeling scenarios. Subsequent models specifically utilize textual data for forecasting stock prices. For instance, [66] customizes distributed vector representations for each company, specifically tailored according to the currently processed textual data. Moreover, [113] computes stock-specific event representations by employing embeddings derived from the NLP representations of each stock ticker, thereby enhancing the predictive precision of temporal financial events. As delineated in Section 2.1, the 'routing by-agreement' method described in [138] facilitates the clustering of feature representations of a stock at a specific time step.

In the study conducted by [91], the model incorporates statistical, time-invariant features that are essential for capturing the fundamental characteristics of the data. Similar to this approach, the embedding representations employed in the proposed ASMs capture essential attributes of the companies under analysis. These characteristics are abstracted and represented within a vector space of the ASM embeddings. Conversely, the model described in [91] utilizes a distinct methodology by integrating what is referred to as 'types of stock' [91] while the ASMs use a more abstract representation in the form of stock embeddings.

In the development of contextual embeddings, an extension beyond the utilization of S2V is proposed through the creation of domain-specific embeddings by aggregating technical indicators into composite embedding vectors (see Section 6.11.2). This methodology is echoed within the existing literature, where various studies incorporate multiple metrics to enrich their models. For example, [237] integrates 152 indicators, while [96], [180] and [74] incorporate 13, 42 and 65 indicators, respectively. However model like [74] are not comparable to the proposed metric usage in the ASMs as they do not take any stock relationships into account. Additionally [65] [5] [210] [162] [238] (which notably includes options and futures), along with [143] [19] [88] (which partly utilizes static features grouped into eight categories), further substantiate the extensive application of technical indicators in enhancing model performance and specificity.

There are select instances in which models from alternative domains are adapted for use with stock data. Some LLM applications integrate non-linguistic data, a less common but growing line of work. In [132], a series of examples are presented in which LLMs are utilized for data from other modalities. Notably, visual transformer models have been adapted for the analysis of time series data relevant to financial markets.

The transformer utilization of different visual models is exemplified in the work of [74]. Here, stock price data is converted into a two-dimensional image format, specifically a 65 × 65 matrix, where each pixel represents one of 65 different indicators for a single stock, albeit without processing inter-stock correlations. Furthermore, an adjacent but notable approach involves leveraging a modified ResNet architecture, traditionally used for image classification, for time series data analysis as documented by [264].

The authors in [270] point out the problem that categorical and textual data can be processed in LLMs, while numerical time series cannot be presented as word embedding. As a solution, an unconventional approach is suggested, as delineated in Section 6.11.3, which involves tokenizing regression data to mimic the entire pipeline of LLMs. According to [102], the work of Xue and Salim [248] presents the first approach to reformulating time series data as text in the form of prompts.

While this strategy is not directly analogous to the core methodology, there exist several precedents where LLMs have been employed for SF and even directly with regression datasets.

Notably [215] [260] as well as [126] demonstrate the integration of LLMs with textual inputs to facilitate SMP. Contrary to traditional fundamental models, these studies sometimes incorporate quantitative data as part of the input set. Specifically, in [260], obfuscated price data are inputted into the GPT-4 [173] model, complemented by additional textual data, enabling the model to generate predictions regarding future market trends. The systematic review in [287], which synthesizes 84 studies (2022–early 2025), categorizes LLM applications in equity markets along two dimensions—end-use cases (e.g., forecasting, sentiment analysis, portfolio management) and technical methods (prompting, fine-tuning, multi-agent systems, RL)—and highlights ongoing challenges in scalability, interpretability, and real-world validation.

Likewise, [126] integrates news and time series data into an LLM, enhancing its predictive capacity. Furthermore, [215] describes a process termed 'Number-to-Text Alignment' [215] where time series features are incorporated into an LLM. Additionally, [111] explores the entry of portfolio weights into an LLM to facilitate adjustments and optimization of investment portfolios. Moreover, [45] employs LLMs for the generation of global event embeddings, a technique which also relates to the integration of extensive global information to address distribution shifts. Also noteworthy is the work presented in [235], where LOB messages are processed by an LLM to generate market data. The LLM employs a specialized vocabulary and tokenizes the numerical values contained within the LOB messages.

The literature points to strong potential for spatio-temporal approaches in SF. It has been argued that transformer models, recognized for their proficiency in modeling relationships and semantic correlations, have not been fully leveraged to address either the temporal dynamics or relational intricacies of stock data—areas in which LLMs possess inherently suitable architectures. Moreover, LLMs exhibit qualities such as few-shot learning and generalizability, which are highly valued in the SF domain, as indicated by existing literature. However, there appears to be a gap in the integration of these attributes within a single model. The authors research aims to bridge this gap by adapting LLMs to fully harness their capabilities for SF.

Research Gap The present thesis introduces several novel contributions that advance the field of quantitative finance through the adaptation of LLMs, the development of foundational models tailored for financial applications, and the formulation of a new, Euclidean, extensible, spatio-temporal modeling framework. A critical gap emerges from the existing body of literature reviewed in this section: the absence of a generalized, extensible, and pretrained model capable of effectively modeling quantitative—or more broadly, time series—data. This model should inherently leverage the advanced spatio-temporal processing capabilities of LLMs originally developed within the NLP domain. Addressing this gap necessitates the resolution of several key research questions, including the design of effective pretraining tasks and an evaluation of their impact on subsequent fine-tuning procedures.

As outlined in [82], a 'Large Investment Model' is pretrained on an extensive dataset yet remains structurally extensible to accommodate diverse financial instruments. In the context of this thesis, each individual training 'sentence'—conceptualized as a sliding window of quantitative financial data—can, in principle, incorporate alternative $\acute{C}^{(t)} \subset C$ and a distinct t at each training step.

The proposed framework represents a significant advancement toward a universal and extensible modeling paradigm. Unlike many non-Euclidean GNNs, which constitute a predominant class of spatio-temporal and patching models, the architecture introduced herein is explicitly designed to overcome their inherent structural limitations.

Fine-tuning of this universal model has been conducted across a range of predictive tasks, examining its adaptability and efficacy. Furthermore, prospective applications in risk assessment and optimization are outlined, illustrating the model's broader utility beyond predictive analytics. A fundamental principle in pretraining is the explicit integration of relationship and intercorrelation information, thereby ensuring that this critical component is not relegated solely to task-specific fine-tuning.

Building on the expected model structure, three key dimensions—correlation information, indicator-based information, and temporal dependencies—are systematically combined into a single model, as proposed in [62]. Importantly, these aspects are incorporated at the pretraining stage, ensuring their foundational role in downstream applications. The direct utilization of LLMs from the NLP domain is a pivotal innovation proposed in this thesis, as these architectures inherently exhibit exceptional spatio-temporal processing capabilities. To the authors knowledge, neither the models, the underlying algorithmic formulation, nor the specific task configurations introduced in this work have been previously explored in the literature. Systematically trained and validated S2V embeddings for downstream use have not been documented previously. Finally, the proposed tokenization strategy, particularly in conjunction with the pretraining paradigm, represents a novel methodological approach that has not been implemented in this manner in existing studies.

3.0.7 Summary of Main Research Gaps

Across the surveyed literature, there is (i) no coherent, pipeline-level treatment of representation learning for multivariate financial time series that is truly analogous to NLP embedding practice, including principled pretraining and systematic downstream validation (3.0.1). (ii) Current Word2Vec adaptations remain methodologically incomplete: they lack clear formulations linked to quantitative targets, thorough CBOW/Skip-Gram comparisons across temporal and spatial axes, and extensive downstream evaluations (3.0.2). (iii) The research gap pertains specifically to the proposed hierarchical model—exemplified by a Clockwork-RNN–style design—whose implementation, ablation strategy, and empirical validation on financial time series remain insufficiently developed; this does not imply that hierarchical approaches in general are underexplored (3.0.3). (iv) Transformer-based approaches show gaps in adapting core pretraining objectives (e.g., multi-axial masking and NSP analogues), handling very long contexts efficiently, preserving cross-asset/channel dependencies beyond naive patching, and offering encoder-only, finance-specific pretraining suited to diverse downstream tasks (3.0.4). (v) Doc2Vec-style market-state embeddings lack dedicated pretraining protocols, robust validation, and demonstrated utility for regularizing rare regime learning (3.0.5). (vi) Finally, there is no spatio-temporal, extensible "foundation" model—here conceptualized as AMSs—that jointly learns relational dynamics end-to-end while enabling few/zero-shot generalization and domain robustness without fixed structural assumptions (3.0.6).

Chapter 4

Methodology and Experiments

In the following, the methods and experiment setup will be explained. The methodological approach combines diverse financial datasets with a focus on leveraging both high-quality U.S. intraday data and broader international data for pretraining discussed in Chapter 5. Training strategies are adapted to account for distribution shifts and model instability, using early stopping and trial-based hyperparameter tuning as explained in Section 4.2. A trading simulation–which is explained in Section 4.3–based on daily buy-hold-sell scenarios is used to assess financial performance under realistic deployment conditions. Section 4.1 describes the research design, the balance between exploratory concept development and hypothesis testing, the evaluation metrics, and the study's methodological position. An overview of the metrics used for the model evaluation and the simulation is given in Section 4.4.

4.1 Research Methodology

A quantitative, experiment-driven research design is adopted, combining two complementary strands. On the one hand, a hypothesis-testing perspective is used, reflected in three guiding research questions and benchmarking against baseline models. On the other hand, an exploratory, methodological path is taken by developing ASMs—a new, extensible model for applying NLP-inspired methods to multivariate financial time series. The latter is presented as the main contribution, while the former provides empirical rigor and testability. Exploratory and confirmatory phases are kept separate: exploratory analyses are used to develop architectures and hypotheses, while confirmatory tests are run on a single time-ordered test set under predefined rules.

Data are evaluated via temporal splits as commonly done in the literature (see Section 2.3), in which each training window strictly precedes its validation and test windows. Any transformation with temporal scope (e.g. normalization) is computed using information available strictly up to the cutoff time (cf. Section 5.2). All data originate from secondary sources, specifically the AV API, and are split into training, validation, and test sets with strict temporal separation to account for non-stationarity. The methodological pipeline is aligned with standard ML practice: data collection is followed by preprocessing and quality checks, then model training, evaluation, and simulation. Reproducibility is ensured through systematic experiment tracking with Weights & Biases [1], version control with GitHub [2], and controlled randomness via seeds in PyTorch[3], in line with practices commonly associated with MLOps.

Two sets of criteria are used. For model development and optimization, SMP accuracy is used as the primary metric, with F1, MCC, and sMAPE as supporting measures to provide a broader view of model behavior. Practical utility in financial contexts is assessed with Sharpe Ratio, IRR, IR, and MD, linking the technical results to realistic investment outcomes (see Section 4.4). In this way, applied ML engineering is integrated with conceptual innovation, and the ASM framework is positioned as both an experimental contribution and a methodological advance.

Two distinct levels of evaluation are pursued. First, predictive models are examined with respect to their performance in SMP/SPP tasks and in trading simulations, providing a direct measure of practical utility. Second, embeddings are analyzed as independent artifacts. Their quality is assessed in two complementary ways: through extrinsic evaluation, where embeddings are implicitly tested

[1] https://wandb.ai/site/

[2] https://github.com/

[3] https://pytorch.org/

by their contribution to downstream predictive performance when integrated into models, and through intrinsic evaluation, where the representational structure of embeddings is studied independently of predictive tasks (cf. Section 7.2.2).

4.2 Training Method

For all models, in addition to the usual hyperparameters, Δt as well as $|C|$ were also tuned as hyperparameters, particularly for the more resource-intensive ASM models.

To determine the optimal number of training epochs, it was approximated via trial and error and then early stopping was used with a patience ψ, after an individual predefined epoch. This approach was necessary because traditional hyperparameter tuning on the validation set (and SGD-based tuning of Θ on the training data) proved to be ineffective. The primary reason for this lies in the differing distributions of the datasets: The training, validation, and test sets follow distinct distributions, making the validation set's performance an unreliable predictor of the expected test set performance. Consequently, a fixed number of epochs was defined for each model. Unless otherwise specified, the SPP/SMP runs were repeated three times per model. The test and validation sets each account for 4% of the total time steps and are separated according to temporary order.

The models exhibit significant instability, which is partly due to the inherent difficulty of forecasting stock prices. As a result, many hyperparameter had to be determined through extensive trial-and-error procedures. All of them are refrained from being listed here. If suitable hyperparameters were identified, a grid search was subsequently conducted in their vicinity (in particular learning rate, ξ, and ρ). Lengthy enumerations are omitted because they do not add to the thesis argument. SMP and SPP were optimized using early stopping based on the best SMP accuracy, as this metric is crucial for determining profit or loss in real-world application scenarios (not MSE or sMAPE).

The training procedures were implemented using the PyTorch framework. Due to the extensive measurement effort and the heterogeneous requirements depending on the model architecture and dataset size, a wide range of hardware configurations was utilized. Specifically, training was conducted on a variety of GPUs available on the HAW Hamburg computing clusters, including 2× H100, 2× L40S, 14× P6000, and 12× V100 GPUs. Depending on the model and dataset, parallel training across multiple GPUs was employed to ensure efficiency. Given the significant variability in execution setups, a detailed listing of the specific hardware allocation per experiment is not provided.

4.3 Simulation

Backtesting and simulations have been introduced in Section 2.3. A simulation technique proposed in [66] is adopted to analyze the practical usage of SF models in real-world trading applications. In particular, a daily trading scenario with a capital of $50K has been simulated, analogous to [66], whereby each day's model prediction is used to buy and hold a selected stock until the market close of the following trading day, at which point the position is sold. Throughout this procedure, no transaction costs are considered, and it is assumed that all orders are executed at the official closing prices (as in [66]). Under this protocol, the cumulative return can be measured by aggregating the daily gains or losses over the complete test horizon. Further key metrics, such as Sharp Ratio, IRR, IR, MDD are reported as was done in [66].

In the present work, the daily buy-hold-sell trading strategy and overall simulation framework from [66] have been applied to assess the performance of SPP-ASM models. Contrasting with [66] all stock predictions generated by the models under study have been incorporated. For the experiments, the best SPP-ASM models identified for each dataset have been selected. Unlike the original evaluation protocol, the entire accumulated capital each day is reinvested instead of using a fixed daily investment amount. The SPP method was favored because its F1 score, as discussed in Chapter 8, was significantly higher than in SMP approaches. Moreover, the ASM variants have been included since they constitute a core contribution of this thesis, offering an extensible, generalizable foundation, particularly under the motivation of research question 3 (see Section 1.2). A schematic representation of the simulation can be seen in Figure 4.1. To assess robustness and generalizability, the simulation was run on both the validation and test splits, and results are reported separately.

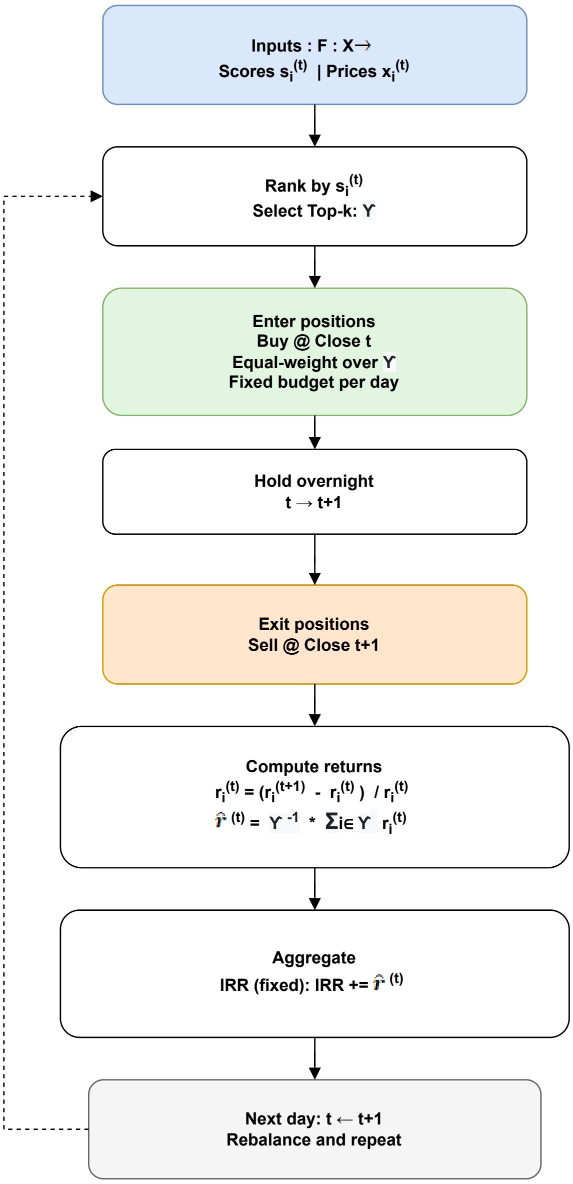


FIGURE 4.1: Schematic representation of the simulation according to [66].

4.4 Metrics

In the following the metrics for the results are defined.

Regression metrics (SPP/SPE)

$$f_{\text{sMAPE}} = \frac{1}{|\mathcal{I}|} \sum_{(i,t)\in\mathcal{I}} 2 \cdot \frac{\left|y_i^{(t+\omega)} - \hat{y}_i^{(t+\omega)}\right|}{|y_i^{(t+\omega)}| + |\hat{y}_i^{(t+\omega)}| + \epsilon} \tag{4.1}$$

$$f_{\text{MAPE}} = \frac{1}{|\mathcal{I}|} \sum_{(i,t)\in\mathcal{I}} \frac{\left|y_i^{(t+\omega)} - \hat{y}_i^{(t+\omega)}\right|}{|y_i^{(t+\omega)}| + \epsilon} \tag{4.2}$$

$$f_{\text{MAE}} = \frac{1}{|\mathcal{I}|} \sum_{(i,t)\in\mathcal{I}} \left|y_i^{(t+\omega)} - \hat{y}_i^{(t+\omega)}\right| \tag{4.3}$$

$$f_{\text{MSE}} = \frac{1}{|\mathcal{I}|} \sum_{(i,t)\in\mathcal{I}} \left(y_i^{(t+\omega)} - \hat{y}_i^{(t+\omega)}\right)^2 \tag{4.4}$$

$$f_{\text{RMSE}} = \sqrt{f_{\text{MSE}}} \tag{4.5}$$

Confusion-matrix definitions Let the ground-truth class be $y_i^{(t)} \in \{0,1\}$ and the predicted class $\hat{y}_i^{(t)} = \mathbb{1}\left(\hat{p}_i^{(t)} \geq \theta\right)$ with threshold $\theta \in (0,1)$ (default 0.5). The *positive class* is the event defining y=1. For SMP in this thesis:

$$y_i^{(t)} = \mathbb{1}\left(x_i^{(t)} > x_i^{(t+\omega)}\right).$$

Using the index set $\mathcal{I}$ (all evaluated (i,t)), define the global (micro-averaged) counts

$$m_{\text{TP}} = \sum_{(i,t)\in\mathcal{I}} \mathbb{1}\left(y_i^{(t)} = 1 \wedge \hat{y}_i^{(t)} = 1\right), \tag{4.6}$$

$$m_{\text{TN}} = \sum_{(i,t)\in\mathcal{I}} \mathbb{1}\left(y_i^{(t)} = 0 \wedge \hat{y}_i^{(t)} = 0\right), \tag{4.7}$$

$$m_{\text{FP}} = \sum_{(i,t)\in\mathcal{I}} \mathbb{1}\left(y_i^{(t)} = 0 \wedge \hat{y}_i^{(t)} = 1\right), \tag{4.8}$$

$$m_{\text{FN}} = \sum_{(i,t)\in\mathcal{I}} \mathbb{1}\left(y_i^{(t)} = 1 \wedge \hat{y}_i^{(t)} = 0\right). \tag{4.9}$$

From these, define the intermediate rates

$$m_{\mathrm{P}} = \frac{m_{\mathrm{TP}}}{m_{\mathrm{TP}} + m_{\mathrm{FP}} + \epsilon} \quad \text{(Precision)}, \tag{4.10}$$

$$m_{\mathrm{R}} = \frac{m_{\mathrm{TP}}}{m_{\mathrm{TP}} + m_{\mathrm{FN}} + \epsilon} \quad \text{(Recall)}. \tag{4.11}$$

Classification metrics (SMP/SMC)

$$f_{\mathrm{Acc}} = \frac{m_{\mathrm{TP}} + m_{\mathrm{TN}}}{m_{\mathrm{TP}} + m_{\mathrm{TN}} + m_{\mathrm{FP}} + m_{\mathrm{FN}}} \tag{4.12}$$

$$f_{\mathrm{F1}} = \frac{2\, m_{\mathrm{P}}\, m_{\mathrm{R}}}{m_{\mathrm{P}} + m_{\mathrm{R}} + \epsilon} \tag{4.13}$$

$$f_{\mathrm{MCC}} = \frac{m_{\mathrm{TP}} \cdot m_{\mathrm{TN}} - m_{\mathrm{FP}} \cdot m_{\mathrm{FN}}}{\sqrt{(m_{\mathrm{TP}} + m_{\mathrm{FP}})(m_{\mathrm{TP}} + m_{\mathrm{FN}})(m_{\mathrm{TN}} + m_{\mathrm{FP}})(m_{\mathrm{TN}} + m_{\mathrm{FN}})} + \epsilon} \tag{4.14}$$

Performance Metrics (Definitions and Prerequisites)

In the following the trading metrics are defined. The have been adapated to follow the notations used in this thesis.

Time, assets, and prices Let $\mathbb{T}$ be the discrete time index and let $|C|$ denote the number of equities. Let $\mathbf{x}_i^{(t)}$ be the close price of asset i at time t.

Post-rebalancing weights and returns Let $m_{\boldsymbol{w}}^{(t)} \in \mathbb{R}^{|C|}$ denote portfolio weights *after* rebalancing at time t, with $\sum_i m_{w,i}^{(t)} = 1$ (fully invested). Define simple per-asset returns

$$m_{r,i}^{(t)} \;=\; \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t-1)}}{\mathbf{x}_i^{(t-1)} + \epsilon}, \qquad m_{\boldsymbol{r}}^{(t)} = \left(m_{r,1}^{(t)}, \ldots, m_{r,|C|}^{(t)}\right)^{\top}.$$

Assuming zero transaction costs, the realized portfolio return is

$$m_r^{(t)} \;=\; \left\langle m_{\boldsymbol{w}}^{(t-1)}, m_{\boldsymbol{r}}^{(t)} \right\rangle.$$

Equity curve Initialize equity with $m_V^{(0)} = V_0$ and update

$$m_V^{(t)} \;=\; m_V^{(t-1)}\left(1 + m_r^{(t)}\right) \quad \forall t \in \mathbb{T}.$$

Equivalently,

$$m_V^{(t)} \;=\; V_0 \cdot \prod_{u \le t} \left(1 + m_r^{(u)}\right) \quad [308].$$

Sample moments and annualization Let

$$\overline{m_r} = \frac{1}{|\mathbb{T}|}\sum_{t\in\mathbb{T}} m_r^{(t)}, \qquad m_{\sigma_r} = \sqrt{\frac{1}{|\mathbb{T}|-1}\sum_{t\in\mathbb{T}}\left(m_r^{(t)} - \overline{m_r}\right)^2}.$$

Let $m_{r_f}^{(t)}$ denote the per-period risk-free return (or a constant r_f) and $\overline{m_{r_f}} = \frac{1}{|\mathbb{T}|}\sum_t m_{r_f}^{(t)}$. Let α be the annualization factor (e.g., α=252 for daily).

Benchmark and active returns. Given benchmark returns $m_{r_b}^{(t)}$, define active returns $m_{r_a}^{(t)} = m_r^{(t)} - m_{r_b}^{(t)}$, with mean $\overline{m_{r_a}}$ and standard deviation $m_{\sigma_{r_a}}$ defined analogously. These quantities are prerequisites for the Information Ratio defined below.

Metrics

Cumulative Return (CR)

$$f_{\mathrm{CR}} \;=\; \frac{m_V^{(\max \mathbb{T})}}{V_0} - 1 \;=\; \prod_{t\in\mathbb{T}}\left(1 + m_r^{(t)}\right) - 1 \quad [308].$$

Sharpe Ratio

$$f_{\mathrm{Sharpe}} \;=\; \sqrt{\alpha}\,\frac{\overline{m_r} - \overline{m_{r_f}}}{m_{\sigma_r} + \epsilon} \quad [309].$$

Information Ratio (IR)

$$f_{\mathrm{IR}} \;=\; \sqrt{\alpha}\,\frac{\overline{m_{r_a}}}{m_{\sigma_{r_a}} + \epsilon} \quad [310].$$

Maximum Drawdown (MDD)

$$m_{\mathrm{peak}}^{(t)} = \max_{s\le t} m_V^{(s)}, \qquad m_{\mathrm{DD}}^{(t)} = 1 - \frac{m_V^{(t)}}{m_{\mathrm{peak}}^{(t)} + \epsilon}, \qquad f_{\mathrm{MDD}} = \max_{t\in\mathbb{T}} m_{\mathrm{DD}}^{(t)} \quad [311].$$

Internal Rate of Return (IRR) Given cash flows $\mathrm{cf}^{(u)}$ at integer offsets $u = 0, \ldots, U$ (positive for inflows, negative for outflows), define

$$m_{\mathrm{NPV}}(\rho) \;=\; \sum_{u=0}^{U} \frac{\mathrm{cf}^{(u)}}{(1+\rho)^u}, \qquad f_{\mathrm{IRR}} \;=\; \left\{\rho \,\middle|\, m_{\mathrm{NPV}}(\rho) = 0\right\} \text{ [312].}$$

Chapter 5

Dataset

The following chapter gives an overview about the dataset used in this thesis. Evaluation in Section 5.2 emphasizes real-world reliability by filtering static data points, avoiding biased weighting, and aligning metrics with practical outcomes.

5.1 Data Acquisition and Market Coverage

The AV API[1] is used for the data. Due to data availability constraints, unfortunately only US-American stocks were able to be meaningfully incorporated, specifically the S&P-500 🇺🇸, into the models for the SMP/SPP tasks. This decision was primarily driven by two factors: First, for non-US markets, access is generally limited to interday data, which rarely extends further back than the early 2010s. This limitation further intensifies the well-known problem of data scarcity, as discussed in Chapter 2.

Second, non-US data is often characterized by substantial gaps, necessitating extensive use of padding methods. In comprehensive experiments conducted on indices such as the CSI300 🇨🇳, DAX-40 🇩🇪, FTSE100 🇬🇧, BOVESPA 🇧🇷, and BSE100 🇮🇳, it is observed that these data limitations rendered meaningful training infeasible under the given conditions, placing such efforts beyond the scope of this thesis. This was reflected, for instance, in an unrealistic above 60% accuracy in interday SMP tasks, which can largely be attributed to the applied padding

[1] https://www.alphavantage.co/

methods, as well as an sMAPE score of 1.180 (compared to a naive baseline score of 1.686). Since intraday data is available for US-American stocks, focusing on this market presents an excellent opportunity to compare the performance, dynamics, and market understanding of the models across different time intervals (i.e. interday data, 60min intraday data and 1min intraday data).

Nevertheless, as outlined earlier, the intercorrelation of stocks constitutes one of the most crucial aspects for accurate predictions, and markets naturally exhibit international interdependencies. Moreover, distinct markets and their respective processes represent valuable knowledge that models may leverage to generalize to analogous situations. Consequently, data from non-US markets was decided to be utilized for the pretraining phase. This approach allows the model to benefit from the insights inherent in these markets. The inferior data quality in these cases is less critical, provided that the downstream performance improves, no systematic distortion of results occurs, and the pretraining — similar to approaches in NLP — is conducted on broader data corpora than the downstream training dataset.

In addition to the S&P-500 🇺🇸, it was decided to include the CSI300 🇨🇳, which is the second most frequently studied index in the literature, as outlined in Chapter 2 (and China is the second biggest economy in the world). Furthermore, the DAX-40 🇩🇪 and the FTSE100 🇬🇧 are included, as these indices are representative of the major stock markets in the home countries of the respective research institutions. The opportunity to examine additional stock markets that fall outside the traditional scope of Western industrialized nations and the OECD 🌐 region would have been welcomed. Such an extension would have been particularly valuable given the presence of spillover effects between various sectors across different nations, which was intended to be investigated in greater detail. However, access to available data is highly dependent on the respective stock exchanges, which has significantly constrained the ability to conduct a more comprehensive analysis in this regard. Stocks corresponding to prominent indices such as the TASI 🇸🇦, IDX 🇮🇩, and Merval 🇦🇷 could not be obtained. An exception to this limitation was the availability of data pertaining to the CSI100 🇨🇳 and CSI300 🇨🇳 indices, which were accessible via the exchanges platforms SSH 🇨🇳,

SHE 🇨🇳, and **SZ** 🇨🇳. It was also found that Brazilian stocks could be obtained from the **SAO** 🇧🇷 exchange, which led to the inclusion of the **BOVESPA** 🇧🇷 index and additional Brazilian stocks in the dataset. Similarly, stocks constituting the **BSE100** 🇮🇳 were retrievable through the **BSE** 🇮🇳. For stocks listed within the **FTSE100** 🇬🇧, data acquisition was achieved by sourcing from the **LON** 🇬🇧 exchange. Correspondingly, data for the **DAX-40** 🇩🇪 was predominantly obtained via the **FRK** 🇩🇪 exchange. In cases where data retrieval proved challenging, fallback options included the **XETRA** 🇪🇺 exchange or the acquisition of ADRs listed on U.S. exchanges. Additional indices incorporated in the dataset included the **CAC40** 🇫🇷 and the **EURO STOXX 50** 🇪🇺, both sourced from the **XETRA** 🇪🇺 exchange. Notably, the latter index comprises stocks originating from various national European indices. With regard to Japanese stocks, data acquisition for the **NIKKEI225** 🇯🇵 and **TOPIX100** 🇯🇵 was constrained. Consequently, the available data from these indices was blended, and their respective stocks were obtained as ADRs. Lastly, ADRs pertaining to the **RTS** 🇷🇺 and the **S&P Africa 40** 🌍 indices were also obtained.

It should be noted that not all stocks from each index were consistently available. This limitation was particularly pronounced for indices where reliance on ADRs was necessary, as only a limited number of such stocks could be obtained. ETF data, which can serve as indicators of various countries' economic performance, were successfully retrieved for multiple nations and tested only on the Baseline models (see Section 7.1 and Appendix A.2).

Furthermore, data for cryptocurrencies and commodities could only be obtained for relatively short periods, typically limited to one year (with interday resolution) or a single trading day (with intraday resolution). This data coverage is insufficient for effectively training an ML model.

The pretraining dataset is defined as

$C = \text{All}^{(2010:)} =$

S&P-500 🇺🇸 ∪ **CSI300** 🇨🇳 ∪ **DAX-40** 🇩🇪 ∪ **FTSE100** 🇬🇧 ∪ **BSE100** 🇮🇳 ∪ **NIKKEI225** 🇯🇵 + **TOPIX** 🇯🇵 ∪ **BOVESPA** 🇧🇷 ∪ **RTS** 🇷🇺 ∪ **CAC40** 🇫🇷 ∪ **EURO STOXX 50** 🇪🇺 ∪ **S&P-40 Africa** 🌍 .

The dataset commences on December 9, 2009, as earlier data was, as previously mentioned, insufficient in quality and quantity. The validation period begins on March 16, 2022, while the test period spans from July 18, 2023, to June 11, 2024. The S&P-500 🇺🇸 dataset starts at the 3rd January 2000 and ends on the June 11, 2024.

5.2 Data Quality and Evaluation Criteria

In the context of intraday data, it is common to encounter instances where the data exhibits no movement. This issue of static values, particularly in the intraday range, is rarely addressed in the literature (with exceptions such as [269]). Fortunately, the distribution of SMP labels for interday data in the dataset is relatively balanced.

Dedicated experiments were conducted to account for the prevalence of non-moving elements in the data, as described in Section 7.7. In these experiments, model performance was evaluated exclusively on the subset of data points that exhibit movement. Consequently, the model was optimized in each step only with respect to these moving elements.

The application of weighted loss functions, such as the approach proposed in [221], proved to be suboptimal for stock market data. This is primarily because the test and validation sets are OOD compared with the training data, and the weighting strategy assumes a distribution that may not hold for these sets. Moreover, utilizing loss weights derived from the test or validation set introduces bias, as such information would be unavailable in a real-world deployment scenario.

For interday data, a threshold was defined to achieve an approximately 50/50 distribution of the labels, following established research approaches outlined in Chapter 2 (e.g., [170]). However, this strategy was not feasible for intraday data due to the presence of non-moving/zero values. The implications of this limitation for model performance are further discussed in Section 8.3. To ensure that the F1-score does **not** misleadingly benefit from the prevalence of non-moving values, the threshold was deliberately adjusted to avoid this effect.

Monotone linear interpolation is used for missing values, which does not affect the SMP labels but is more stable for SPP and also more accurate in the evaluation. Further data cleaning and preparation information can be found in Appendix A.6.

Chapter 6

Proposed Adapted Approaches for Stock Forecasting

In the following, the proposed models and the methodological framework adopted from the domain of NLP for SF are delineated. The methodology is aligned with the standard NLP pipeline, with targeted deviations to test the effectiveness and utility of specific components for SF. This pipeline structure is discussed in Section 6.1.

6.1 Conceptual Architecture of LLMs

The architecture of an LLM within the NLP paradigm was previously elucidated in [222]. For coherence, the conceptual framework is briefly restated here. These frameworks are theoretical; practical implementations may vary, especially in training procedures such as W2V.

The pertinent stages of the NLP pipeline, along with their proposed adaptations to the specific components of the LLM framework, are depicted in Figure 6.1.

The word token embedding model $\tilde{F}^{\text{<E>}}\left(\left(\tilde{v}^{(i)}\right)_{1=i}^{\tilde{l}}, \tilde{E}\right)$ constructs the embeddings $\left(\tilde{e}^{(i)}\right)_{1=i}^{\tilde{l}}$ utilizing a (pretrained) embedding matrix $\tilde{E}$. The input to $\tilde{F}^{\text{<E>}}(.)$ comprises the word tokens $\left(\tilde{v}^{(1)}, \tilde{v}^{(2)}, \ldots, \tilde{v}^{(\tilde{l})}\right) : \forall \tilde{v}^{(i)} \in \tilde{V} \subset \mathbb{N}$. These tokens are generated by a tokenizer $F^{\text{<TO>}}(\tilde{X}, \tilde{V})$ utilizing the input text $\tilde{X}$ and the predefined vocabulary $\tilde{V}$.

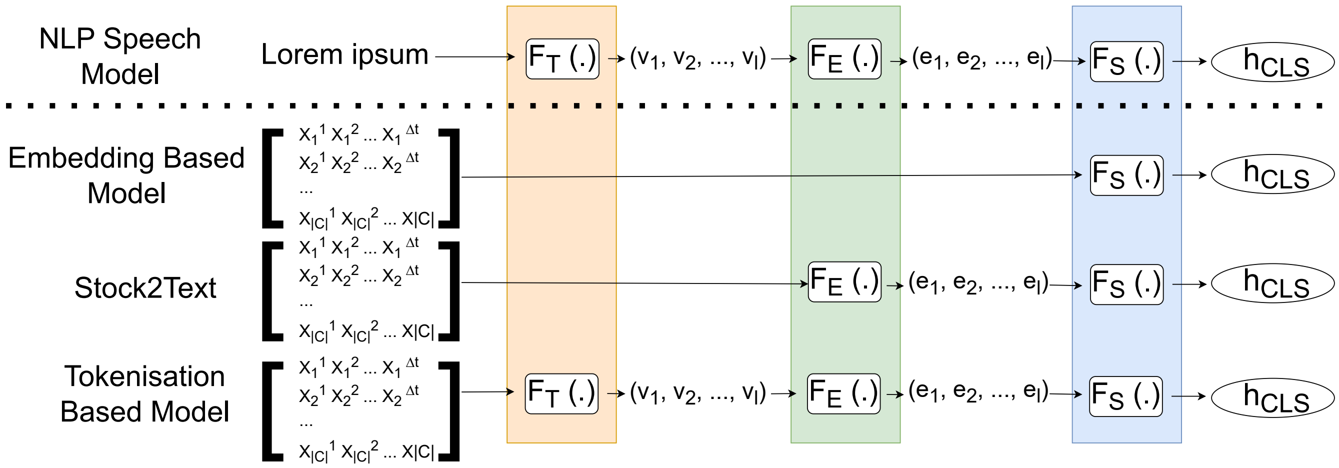


FIGURE 6.1: NLP/ASM Pipeline build-up visualization. The illustration is taken from the authors publication [222] and modified.

In the realm of predictive LLMs, the classifier token $\tilde{\mathbf{h}}_{\text{CLS}}$ is integrated into a prediction head, for instance, a classification head, to facilitate the execution of specific model tasks, such as sentiment analysis. Generative LLMs are not considered; the discussion is restricted to predictive approaches. As elucidated in Chapter 2, predictive models represent the predominant trend in SF, attributed to the inherent complexity of SF.

6.2 Notations

This chapter introduces the notation used throughout the thesis to ensure a consistent and unambiguous formulation of the models, methods, and mathematical concepts. No claim of originality is made for the notation; common conventions are followed, primarily aligned with [43], which is widely used in SF.

Basic Notations Each tensor is denoted by Latin letters, where any tensor X with $\text{rank}(X) \leq 2$ is represented by uppercase letters, while tensors with $\text{rank}(\mathbf{x}) = 1$ are represented in bold lowercase form. Sets are similarly expressed in uppercase Latin letters, e.g., M. Functions are denoted by lowercase letters, such as $f(.)$. When a function represents an ML model, it is expressed in uppercase, e.g., $F(.)$. The model's name is indicated as a superscript above, $F^{\langle\text{name}\rangle}$. Scalars are represented by lowercase Latin letters, e.g. a. Hyperparameters are expressed using lowercase Greek letters, such as λ. Abstract concepts are represented by uppercase Greek letters or with special formatting, such as $\mathbb{T}$. These include, for example, contextual information for predictions Π or general model parameters Θ. The $\tilde{}$ symbol is used when discussing NLP concepts to distinguish them more easily from topics in stock forecasting. Since a spatio-temporal problem is addressed, the time axis is consistently represented using superscripts and the spatial axis using subscripts. For example, the temporal axis is denoted as $X^{(t)}$ and the spatial axis as X_i. Furthermore, consistent notation is maintained for the same concepts throughout all chapters, as defined in the following.

Data The entire period of a stock price is considered as $\mathbb{T}$. Stock prices are denoted for stocks $c_i \in C$, where C is the set of all assets within the market context currently used. Stock prices, even for a single company, are typically not provided as univariate time series but rather as a feature vector comprising interval-based features such as Open price, Close price, High price, Low price, and trading Volume (OHCLV [70] [144]). Additional features may also be included. The price feature of a stock c_i at time t is denoted as $\mathbf{x}_i^{(t)} \in \mathbb{R}^{\mathbb{F}}$. Brackets are used around the superscript to distinguish it from an exponent when referring to the timestep.

The horizon observed by a model (also referred to as the lookback window) is denoted as Δt. Consequently, the complete dataset is $\hat{X} \in \mathbb{R}^{|C| \times \mathbb{T} \times \mathbb{F}}$.

For most models, stacked representations $\grave{X} \in \mathbb{R}^{(|C| \cdot \mathbb{F}) \times \mathbb{T}}$ are utilized, which are defined as

$$\forall c_i \in C, \forall t \in \mathbb{N} < \mathbb{T}, \forall f \in \mathbb{N} < \mathbb{F} : k = (i-1) \cdot \mathbb{F} + f \tag{6.1}$$

and $\grave{X}[k,t] = \hat{X}[i,t,f]$.

To simplify the notation for indexing data access $x_i^{(t)} = X[i,t]$, $X_i = X[i]$ and $X^{(t)} = X[j,t]$ with $1 \leq j \leq |C| \cdot \mathbb{F}$ is defined. At each training step, $i \sim \mathcal{U}(\mathbb{N} < \mathbb{T} - (\Delta t + \omega))$ and $X[v,j] = \grave{X}[v, i+j]$, $\forall j \in \mathbb{N} < \Delta t$ are defined. Thus, the model receives an input $X \in \mathbb{R}^{(|C| \cdot \mathbb{F}) \times \Delta t}$.

In practice, instead of using raw data, we typically utilize returns, relative returns, or (relative) log returns. A return (defined here for one price feature) is given by $x_i^{(t)} - x_i^{(t-1)}$ a relative return [50] is defined as $\frac{x_i^{(t)} - x_i^{(t-1)}}{x_i^{(t-1)}}$, a log return as [196] $\log(x_i^{(t)}) - \log(x_i^{(t-1)})$ and a relative log return as $\frac{\log(x_i^{(t)}+\epsilon) - \log(x_i^{(t-1)}+\epsilon)+1}{\log(x_i^{(t-1)}+\epsilon)+1}$. To the authors' knowledge, this specific definition of a relative log return has not been previously published; independent prior use cannot be ruled out.

Throughout, we denote returns, log returns, or relative returns consistently as X, unless explicitly specified otherwise. After embedding through a latent layer $F^{\langle \mathrm{LL} \rangle}$, X is in the embedding space and is represented as $\bar{X} \in \mathbb{R}^{\xi \times \Delta t}$. Conceptually, this latent transformation serves a function analogous to embeddings in NLP: each market snapshot is projected into a structured representation, ensuring that its positioning within the vector space is meaningful relative to other snapshots. Therefore we can regard the latent layer as analogous to the W2V embedding matrix, which functions not for indexed word tokens but for regressive market snapshot data; $X^{(t)} \equiv \tilde{v}^{(t)} \Rightarrow F^{\langle \mathrm{LL} \rangle} \equiv \tilde{E}$.

The author empirically verified that utilizing returns, relative returns, or log returns yields better results than normalization, except for Volume (or other sizeable feature values), which would become excessively large otherwise. If normalization is applied, it is conducted on a feature/channel basis, as exemplified in [233], due to the significant differences in the magnitudes of feature value ranges.

Machine Learning In general, the focus is placed almost exclusively on time series-based stock price prediction with the objective of forecasting prices at $t+1$. Accordingly, an ML model can be expressed as $F_\Theta : X \mapsto \hat{\mathbf{y}}$. In most cases, $\hat{\mathbf{y}} \in \mathbb{R}^{|C|}$ holds.

Within the scope of SF tasks, a distinction is made between SPP and SMP. The correct values, i.e., the labels, are denoted by $\mathbf{y}$. The goal of SPP is to predict $\mathbf{y} = X^{(t+\omega)}$. For SMP, the target is

$$\mathbf{y} = \mathbb{I}^{(t)}(X^{(t)} > X^{(t+\omega)}) \tag{6.2}$$

predicting a binary label of 0 or 1 to identify decreases or increases (notation follows Yoo et al. [258]). The offset for the prediction target is ω, which is typically set to $\omega = 1$. SMC is defined as SMP with $\omega = 0$ and SPE is SPP with $\omega = 0$ where $\mathbf{y} \notin \mathbb{R}^{|C|}$ holds and details of the task can be found in the corresponding model descriptions.

The CLS token $\mathbf{h}_{\text{CLS}} \in \mathbb{R}^{\xi}$ typically represents the model's condensed understanding of input sequences and serves as the basis for the final classification in most cases. The tensor H is the model's last hidden state, and unless otherwise specified, $H \in \mathbb{R}^{\xi \times \Delta t}$ holds. The layer number is denoted by ρ. In iteration-based models, such as recurrent models, the iteration number is referred to as κ.

The final classification in most models is performed using $\hat{\mathbf{y}} = F^{\langle \text{CLS} \rangle}(\mathbf{h}_{\text{CLS}})$, where $F^{\langle \text{CLS} \rangle}$ is a linear layer with an initial tanh(.) activation function and batch normalization. For SPP/SPE no activation function is used and for SMP/SMC the logits are calculated i.e. the σ function is used. The notation $\mathcal{P}()$ denotes probabilities and $\mathcal{X}$ the random variable i.e $\mathcal{P}(\mathcal{X} = x)$. A vector containing only ones is denoted by $\mathbf{1}$.

Repeatedly used Parameters Numerous parameters and notations are employed across various models, which are enumerated here. In principle, these parameters are redefined in each chapter, and specific indices are used when referring to parameters from a previous chapter. In most models, the model size or hidden size is denoted as ξ. The loss function is defined as $\mathcal{L}$. Parameters λ

represent balancing parameters. Thresholds are denoted by θ, and sliding window sizes are indicated by ϖ.

Tables In tables, upward arrows next to a metric's name signify that higher values indicate better performance, and vice versa. Regression metrics f_m are usually calculated as $f_m(y_i^{(t)}, \hat{y}_i^{(t)})$. In order to provide a comparison or baseline for the regression values, the mean value of all $f_m(x_i^{(t)}, x_i^{(t+1)})$ is indicated as a naive model. The information for the test sets and validation sets is reported separately, as the values sometimes deviate significantly from each other due to the non-stationarity of the time series. The validation set and test set performance is reported for each metric. The best accuracy is highlighted by writing them in bold.

Rest The sigmoid function is represented as $\sigma(.)$, and binary cross-entropy is denoted by $\mathcal{H}$. The symbol $\odot$ is used to denote the concatenation of two tensors along the second axis, while $\otimes$ represents the Hadamard product. A logical xor is represented as $\oplus$.

The $\overline{\underline{\bullet}}$-function is defined as the multiplicative counterpart to the $\text{abs}(.) = |.|$ function. To denote this, $\overline{\underline{x}}$ is written, where the lines represents the analogy to the absolute value notation $\text{abs}(x)$ or $|x|$.

The function is defined as:

$$\overline{\underline{x}} = \exp|\ln x| \; . \tag{6.3}$$

To the best of the authors knowledge, this function has not been previously introduced in the literature. Assignments / redefinitions of variables are marked with $\leftarrow$ or $\rightarrow$. In contrast, logical implications are denoted by $\Rightarrow$ or $\Leftarrow$. The Cartesian product of sets A and B is denoted by $A \bowtie B$, to distinguish it from tensor axis dimension size definitions ($\times$). The $\mapsto$ arrow is used for function mapping. $A \trianglelefteq B$ is used to indicate that tensor A is fully contained within tensor B.

6.3 Preliminaries

Before the model details, key assumptions and notational conventions are specified for clarity. This discourse predominantly addresses the TS prediction of stock prices, framing SF as a multivariate time series forecasting endeavor. Fundamental information (e.g., text from social media, reports, or news) is intentionally left out of the analysis. Instead, the informational inputs used consist solely of numerical time series indicators, including several technical indicators, together with data on the industries and sectors linked to the stocks (TSG).

While the primary focus remains on stock prices, it is pertinent to mention that within the context of this thesis, the term 'stock' is used interchangeably to refer also to ETFs and other financial assets. However, the scope of this thesis does not extend to other asset classes such as derivatives, currencies, currency exchange rates, commodities, and the like.

For the majority of models, with the notable exceptions of S2V and AMS, market information at t is represented as $X^{(t)}$, consisting of stacked feature vectors. These vectors may include OHCLV features, other technical indicators, or scaled features specific to S2V. This representation enables the model to encapsulate both inter-stock and intra-stock correlations within a temporal context. This has been identified by Li et al. [124] as a pivotal approach for SF models.

In the proposed methodology, each market snapshot is conceptualized as a 'token' in NLP, similar to Li et al. [124]. This conceptual alignment allows the temporal axis of stock trends to be paralleled with the positional indices used in text sequences within NLP. Furthermore, the dimensions of stock features i.e the spatial axis is mapped onto the embedding dimension in NLP, i.e. $|C| \equiv \tilde{\xi} \wedge \Delta t \equiv \tilde{l}$.

For the fine-tuning of all models, SPP and SMP are specifically focused on as downstream tasks, with or without pretraining, depending on the model configuration. These tasks involve generating alpha predictions as defined in [82]. Consequently, the models are primarily applicable to directional trading and long-short trading strategies, as defined in [82]. Outputs such as position recommendations or portfolio rebalancing fall outside the scope of this thesis. Additional downstream applications, including their potential utility in risk management and market simulation, are discussed within the context of ASMs in Section 9.1.

6.4 Basic Modules

In the following the basic modules are introduced.

Information Embedding To enhance model training, additional information is incorporated by introducing learnable embeddings, a technique commonly employed in NLP. This approach, similar to the inclusion of positional embeddings, allows the model to encode auxiliary features, potentially improving its ability to capture complex patterns and dependencies within the data. This approach is frequently adopted in multimodal NLP research, particularly in V+L models, as for example demonstrated in [23]. Especially with stacked feature inputs of the form

$$X \in \mathbb{R}^{(|C|\cdot\mathbb{F})\times\Delta t} \tag{6.4}$$

this can be useful. For readability, $X_{i,d}^{(t)}$ is defined as the feature d of c_i in timestep t in X. First, the stock embeddings are defined as a learnable matrix $E_C \in \mathbb{R}^{|C|\times\Delta t}$. Further

$$X_{i,d}^{(t)} \leftarrow X_{i,d}^{(t)} + E_C[i,t] \tag{6.5}$$

is assigned. Furthermore the interval feature embedding matrix $E_{\mathbb{F}} \in \mathbb{R}^{\mathbb{F}\times\Delta t}$ and

$$X_{i,d}^{(t)} \leftarrow X_{i,d}^{(t)} + E_{\mathbb{F}}[d,t] \tag{6.6}$$

is defined.

Incorporating additional sector-specific information regarding each company may enhance the model's predictive performance. Stocks listed on exchanges can be organized into various sector classifications. One such taxonomy, derived from the AV database, groups stocks into eight primary sectors: technology 💡, trade & services 🛒, manufacturing 🏭, finance 🪙, life sciences 🧬, energy & transportation ⚡, real estate & construction 🏠 and, other (null) ∅. During this thesis, these symbols are appended to stock tickers to indicate the corresponding sector, thereby facilitating the reader's understanding and classification of the respective stock. To formalize this categorization, a set of sector indices is defined as $K = \{k_1, \ldots, k_{|K|}\}$, where each index $k_j \in \mathbb{N}$ denotes the unique identifier for a specific sector category. An auxiliary vector $\mathbf{k} \in \mathbb{R}^{|C|}$ is introduced to indicate the sector category for each stock.

Further, the option of utilizing industry-specific information is available (analogously defined as $\dot{K}$). However, this data is employed selectively across various methodologies due to its highly specialized nature. Within the S&P-500 🇺🇸 dataset, for instance, there are 191 distinct industries, 92 of which appear only once, significantly limiting the potential for the models to generalize patterns of specific industries.

The sector-specific embedding matrix $E_K \in \mathbb{R}^{8 \times \Delta t}$ which captures the distinct learnable features associated with each of the eight sectors is defined. The incorporation of sector information is realized by augmenting the stock time series as

$$X[i,j] \leftarrow X[i,j] + E_K[\mathbf{k}[i \mod \mathbb{F}], j] \tag{6.7}$$

to embed sector-specific characteristics within the representation. The resulting representation can be used across architectures, including the proposed transformers, during pretraining and fine-tuning to incorporate sector information.

Transformer-based models require additional contextual information within the attention mechanism to accurately differentiate individual positions within embeddings [219]. To address this, relative positional embeddings are utilized, as proposed in [205], [171] (rel-e, rel-b, and rel-b), and [31]. Furthermore, rotary position embeddings from [212] are tested. For comprehensive implementation details, interested readers are directed to the original publications.

Handling Non-Stationarity As extensively discussed in Chapter 2, stock data is inherently non-stationary. This characteristic poses substantial challenges, particularly when there is a significant temporal gap between the training set, validation set, and test set. In practice, distributional shifts are observed not only for prices but also for returns, relative returns, and (relative) log-returns. Consequently, regularization of X is required at each timestep to maintain stability in the model's predictions.

In this approach, an adaptation of the method presented by [263] is experimented with, as it has demonstrated promising results in addressing non-stationarity. To mitigate the non-stationarity within stock data, a modified 'Normalization Module' from [141] is implemented, while the 'De-stationary attention' is omitted. This proposed methodology aligns with the approach taken by [33], providing a reliable baseline for handling temporal variance in financial time series data. Furthermore, for the ASM in Section 6.11, it has been found that the method described in Appendix A.1 performs better.

6.5 Proposed Baseline Models

Chapter 2 outlines the datasets and the difficulties in comparing SPP and SMP methods reported in prior work. Although alignment with prior models is pursued, the main goal is to establish a baseline that clarifies the relative performance of the proposed models. Also the baseline-experiments are intended to expose the main challenges of SPP and SMP within a simplified setting. For both SMP and SPP, three simple baseline models $F^{\langle \text{BM} \rangle} : X \mapsto \hat{\mathbf{y}}$ are defined. A multi-layer transformer $F^{\langle \text{BM-T} \rangle}$, a multi-layer RNN $F^{\langle \text{BM-R} \rangle}$, and a multi-layer LSTM $F^{\langle \text{BM-L} \rangle}$ are chosen. RNN and LSTM models are selected, as they are commonly employed as baseline models for SF, as demonstrated in [286]. The primary objective of the baseline models is to enable comparison and evaluation of the effectiveness of the developed models in this thesis, as the heterogeneity of the data sets identified in Chapter 2 otherwise prevents comparison with other models. For the proposed $F^{\langle \text{BM-T} \rangle}$ the concept of [263] is followed, H is flatted and the size of the prediction layer is increased accordingly. Transformer models are specifically introduced to assess the criticisms raised in [261] and [272] regarding their application to time series processing, enabling the more complex (transformer-based) models implemented later to be evaluated from these perspectives.

6.6 Proposed Stock2Vec Models

This section is mainly based on the authors publications [220] and [224].

The initial stage in the NLP pipeline involves constructing of $\tilde{E} \in \mathbb{R}^{|V| \times \tilde{\xi}}$. Notably, SG and CBOW are the predominant methodologies for training the embeddings $\tilde{e}^{(t)}$. The foundational concept of representing relationships between word tokens in NLP revolves around these word token embeddings.

Expanding on this concept, as proposed in Chapter 1, an analogous approach can be applied to financial markets, where stock entities are transformed into high-dimensional embeddings. This approaches facilitates the expression of inter-stock relationships and correlations. The core idea of the S2V models is to use spatial associations among stocks represented as high-dimensional embeddings. As discussed in Chapter 2, forecasting stock prices is inherently stochastic and highly sensitive to volatility. Nevertheless, representing spatial interconnections effectively may support methods that predict the consequences of unforeseen market events (though not the events themselves). Furthermore, a model that is good at recognizing the relationships between stocks can rely on the prediction of individual stocks (possibly more predictable ones) and infer the performance of others from this relationship information. In particular, assuming market efficiency is not absolute—especially at high-frequency intervals—this approach may provide a predictive advantage.

Section 3.0.6 discussed the prevalent use of predefined relational information within embeddings. However, an emerging body of literature advocates for a dynamic training approach. Focusing on time series analysis, this research will exclusively utilize stock price data to construct these embeddings. Seo et al. [203] have emphasized that time series data, in adherence to the EMH, suffices for representing relationships as it ostensibly encompasses all necessary information.

Unlike the methodologies that employ GNNs as described by Seo et al., the W2V models are adapted as the proposed S2V algorithms in this study to better suit the data-driven objectives. This adaptation is predicated on the hope that the well-established concepts from the NLP domain will prove efficacious in quantitative finance as well.

Brief Review of W2V in NLP The prediction of an SG model $\tilde{F}^{\langle \text{SG} \rangle}(.)$ can be formally described as

$$\tilde{F}^{\langle \text{SG} \rangle}(\tilde{v}^{(t)}) = f_{\text{softmax}_j}(\tilde{E}[\tilde{v}^{(t)}] \cdot W_{\text{SG}}^T + \mathbf{b}_{\text{SG}}) \tag{6.8}$$

with $W_{\text{SG}} \in \mathbb{R}^{(2 \cdot \tilde{\varpi} \cdot |\tilde{V}|) \times \tilde{\xi}}$ and the $f_{\text{softmax}_j}(.)$ function being applied to each of the $2 \cdot \tilde{\varpi}$ sliding window predictions [156].

The resulting $|\tilde{V}|$ dimensional stacked vectors represent the probability for each $\tilde{v}^{(t)} \in \tilde{V}$ to appear in the sliding window context $\{\tilde{v}^{(t-\tilde{\varpi})}, \tilde{v}^{(t-(\tilde{\varpi}-1))}, \ldots, \tilde{v}^{(t-1)}, \tilde{v}^{(t+1)}, \ldots, \tilde{v}^{(t+(\tilde{\varpi}-1))}, \tilde{v}^{(t+\tilde{\varpi})}\}$ of the current word token $\tilde{v}^{(t)}$. For $\mathcal{L}_{\text{SG}}$ the mean Cross-Entropy is calculated between the predicted sliding window probabilities and the true word token context.

Conversely, for the CBOW approach, the training of $\tilde{F}^{\langle \text{CBOW} \rangle}(.)$ can be defined as

$$\begin{aligned}\tilde{F}^{\langle \text{CBOW} \rangle}((\tilde{v}^{(t-\tilde{\varpi})}, \tilde{v}^{(t-(\tilde{\varpi}-1))}, \ldots, \tilde{v}^{(t-1)}, \tilde{v}^{(t+1)}, \ldots, \tilde{v}^{(t+(\tilde{\varpi}-1))}, \tilde{v}^{(t+\tilde{\varpi})})\ ) \\ = f_{\text{softmax}}\left(\left(\sum_{j=t-\tilde{\varpi}}^{(t+\tilde{\varpi})} \tilde{E}[\tilde{v}^{(j)}] \right) \cdot W_{\text{CBOW}}^T + \mathbf{b}_{\text{CBOW}} \right)\end{aligned} \tag{6.9}$$

with $W_{\text{CBOW}} \in \mathbb{R}^{|V| \times \tilde{\xi}}$ [156].

While actual implementations of both SG and CBOW might differ and other more efficient approaches such as [181] exist, the above is sufficient as an intuition to transfer the model into the SF.

Adaption in the Stock Domain In the SF domain, the feature vectors derived from OHCLV data encounter several challenges when adapted to NLP applications. Firstly, these vectors are characterized by their low dimensionality, which may limit their effectiveness in capturing the complex patterns typically required in NLP models. Additionally, the feature vector associated with a particular stock c_i, fails to encapsulate any information regarding its relational dynamics with other stocks $c_{j \neq i}$. This lack of relational information limits their applicability in NLP models, where representing interdependencies/spatial information is essential.

It would be advantageous to transform stock prices into high-dimensional vectors i.e. representations, wherein positions within the vector space hold domain-specific significance. Whereas the desired vector positions in a W2V model correlate directly with word tokens, adapting this approach to the S2V framework presents non-trivial challenges due to the stochasticity of the market in the dynamic, time-varying relationships of stocks [48].

Viewing stock data as a multivariate time series allows for the redefinition of context within SG and CBOW models along both the temporal and the 'Market'-axis (spatial axis). While the temporal axis commonly receives primary attention in SF, such as in univariate stock price prediction, [257] also incorporates the market axis in its pretraining methodology.

To redefine the adaptation along the temporal axis, the proposed SG methodology necessitates employing a specific stock price feature $x_i^{(t)}$, to predict adjacent stock prices within the set $\{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\} \setminus \{x_i^{(t)}\}$. Conversely, the CBOW approach, when adapted to a CBOS model, utilizes the context $\{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\} \setminus \{x_i^{(t)}\}$ to predict the stock price $x_i^{(t)}$.

Both methodologies are evaluated in the subsequent sections. It is imperative to acknowledge that SPP constitutes a regression task, characterized by the relative continuity of stock prices and the rarity of substantial discrepancies and cases where $|x_i^{(t+1)} - x_i^{(t)}| \gg 0$ holds. This contrasts with NLP, where the predicted values $\tilde{v}^{(t+1)}$, can vary widely since v represents merely an index within V.

In this analysis, methods predicated on predicting the prices of identical stocks, i.e. operating on the temporal axis, are distinctly identified with the label 'X' (denoted as $F^{\langle \text{X-SG} \rangle}(.)$ and $F^{\langle \text{X-CBOS} \rangle}(.)$) to signify a focus on one specific stock price, x. To represent $x_i^{(t)}$, any of the $\mathbb{F}$ features may be employed. It is common in related research to utilize the Close price as a representative feature.

The usage of $\mathbf{x}$ is also possible by feeding either $\mathbf{x}_i^{(t)}$ or $\{\mathbf{x}_i^{(t-\varpi)}, \ldots, \mathbf{x}_i^{(t+\varpi)}\} \setminus \{\mathbf{x}_i^{(t)}\}$ to the model and predict either $\mathbf{x}_i^{(t)}$ or $\{\mathbf{x}_i^{(t-\varpi)}, \ldots, \mathbf{x}_i^{(t+\varpi)}\}$. Models using all $\mathbb{F}$ interval features are marked with '$\mathbb{F}$'.

To enhance the understanding of inter-stock relationships, a more robust methodology is proposed, which utilizes the Market Snapshot $X^{(t)}$ (operating on the spatial/market axis). Notably, the scalar $x_i^{(t)}$, representing any price feature, could serve as a predictive tool for the price of another company $c_{j \neq i}$. This approach enables the embedding vector to be contextualized with respect to the stock axis, rather than focusing only on the univariate time series components of X_i.

Proposed techniques, operating on the spatial axis, necessitating the prediction of prices from other stocks are denoted with 'C' ($F^{\langle \text{C-SG} \rangle}(.)$ and $F^{\langle \text{C-CBOS} \rangle}(.)$), referencing the set of companies C. Furthermore, the vector $\mathbf{x}_i^{(t)}$ can be used as input to the model, which aims to predict the vectors $\mathbf{X}^{(t)}$ for all stocks. Models employing this strategy are subsequently identified with the symbol $\mathbb{F}$ throughout the subsequent sections. The proposed approach presented at [220] is followed to incorporate current stock price data into the embedding vectors. This integration is achieved through a scaling operation denoted by $\star$ (as in [222]), which can manifest either as multiplication or addition. For clarity in subsequent discussions, the term 'stock price', represented by $x_i^{(t)}$, will be employed as a proxy for any arbitrary OHCLV feature associated with the stock or all of them. The definitions are first done for SPE and in Section 6.6 a concept is proposed to use all of them for SMC.

Skip-Gram Adaption Initially, the proposed modifications applied to the SG models are introduced.

C-SG Model The initial training objective is to predict the present values of all $x_{j \neq i}^{(t)}$ given the current stock price of one specific company $x_i^{(t)}$. To achieve this, the model $F^{\langle \text{C-SG} \rangle}(.)$ is utilized, which is formally defined as

$$F^{\langle \text{C-SG} \rangle}(x_i^{(t)}) = (E[i] \star x_i^{(t)}) \cdot W_{\text{C-SG}}^T + \mathbf{b}_{\text{C-SG}} \tag{6.10}$$

, with $W_{\text{C-SG}} \in \mathbb{R}^{|C| \times \xi}$. The output from this model is a vector of dimension $|C|$ encapsulating the predicted current stock prices $x^{(t)}$ of all $c_j \in C$ (including c_i).

C-Alt-SG Model A refined approach to the implementation of $F^{\langle \text{C-SG} \rangle}$ involves excluding the current c_i from representation in $\hat{\mathbf{y}}$. This approach should force the model to recognize the stock under consideration and adjust the prediction accordingly. This objective is achieved through the following adjustment of the model:

$$F^{\langle \text{C-Alt-SG} \rangle}(x_i^{(t)}) = (E[i] \star x_i^{(t)}) \cdot W_{\text{C-Alt-SG}}^T + \mathbf{b}_{\text{C-Alt-SG}} \tag{6.11}$$

with $W_{\text{C-Alt-SG}} \in \mathbb{R}^{(|C|-1)\times\xi}$. The standard practice of averaging the embeddings E and the matrix $W_{\text{C-Alt-SG}}$, as commonly employed in NLP embedding training methodologies [227], becomes impossible due to the resultant dimensions.

$\mathbb{F}$-C-SG Model To use $\mathbf{x}$ the model $F^{\langle \mathbb{F}\text{-C-SG} \rangle}(x_i^{(t)})$ is defined as

$$F^{\langle \mathbb{F}\text{-C-SG} \rangle}(\mathbf{x}_i^{(t)}) = \begin{bmatrix} E[i] \star \mathbf{x}_i^{(t)}[1] \\ \dots \\ E[i] \star \mathbf{x}_i^{(t)}[\mathbb{F}-1] \end{bmatrix} \star W_{\mathbb{F}\text{-C-SG}}^T + \mathbf{b}_{\mathbb{F}\text{-C-SG}} \tag{6.12}$$

with $W_{\mathbb{F}\text{-C-SG}} \in \mathbb{R}^{|C|\cdot\mathbb{F}\times\xi\cdot\mathbb{F}}$.

The definition for the alternate function mirrors the previously stated model, yet it employs $W_{\mathbb{F}-\text{Alt-C-SG}} \in \mathbb{R}^{(|C|-1)\cdot\mathbb{F}\times\xi\cdot\mathbb{F}}$.

X-SG Model The second training objective within the S2V SG framework involves the estimation of future and past stock prices based on a current observation, operating on the univariate temporal axis. Specifically, given the current stock price $x_i^{(t)}$, the model is tasked with predicting the context $\{x_i^{(t-\varpi)}, x_i^{(t-(\varpi-1))}, \dots, x_i^{(t-1)}, x_i^{(t+1)}, \dots, x_i^{(t+(\varpi-1))}, x_i^{(t+\varpi)}\}$. This prediction task is a direct adaption of $\tilde{F}^{\langle \text{SG} \rangle}(.)$ as $F^{\langle \text{X-SG} \rangle}$. The model $F^{\langle \text{X-SG} \rangle}(.)$ for this adaptation is defined as

$$F^{\langle \text{X-SG} \rangle}(x_i^{(t)}) = (E[i] \star x_i^{(t)}) \cdot W_{\text{X-SG}}^T + \mathbf{b}_{\text{X-SG}} \tag{6.13}$$

where $W_{\text{X-SG}} \in \mathbb{R}^{(2\cdot\varpi)\times\xi}$ holds true. This operation projects $x_i^{(t)}$ into a $2\cdot\varpi$ dimensional vector space.

$\mathbb{F}$-X-SG Model Again an alternative model can be defined which allows for the comprehensive utilization of the entire feature vector $\mathbf{x}_i^{(t)}$. The model

$F^{\langle \mathbb{F}\text{-X-SG} \rangle}(\mathbf{x}_i^{(t)})$ is articulated as

$$F^{\langle \mathbb{F}\text{-X-SG} \rangle}(\mathbf{x}_i^{(t)}) = \begin{bmatrix} E[i] \star \mathbf{x}_i^{(t)}[1] \\ \ldots \\ E[i] \star \mathbf{x}_i^{(t)}[\mathbb{F}] \end{bmatrix} \cdot W^{T}_{\mathbb{F}\text{-X-SG}} + \mathbf{b}_{\mathbb{F}\text{-X-SG}} \tag{6.14}$$

where $W_{\mathbb{F}\text{-X-SG}} \in \mathbb{R}^{(2\cdot\varpi\cdot\mathbb{F})\times\xi\cdot\mathbb{F}}$ holds. Furthermore, the models can be combined by summing their loss terms, allowing the embedding E to be trained jointly across multiple tasks. This integration is quantified by the loss term $\mathcal{L}_{\text{S2V-SG}}$.

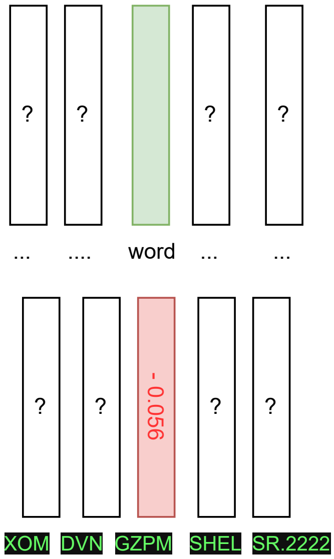


FIGURE 6.2: Schematic sketch of the C-SG approach.

An illustration of the C-SG principle is provided in Figure 6.2. Similar to the C-CBOS approach, the C-SG framework maps the current price information of a stock to a word in the NLP-SG paradigm. In the NLP setting, context windows are inherently dynamic, as each position is occupied by different words drawn from natural text which the sliding window iterates over. In contrast, within the financial adaptation, the set of stocks serving as context i.e. C remains fixed (e.g., the constituents of the S&P 500 🇺🇸), with numerical differences arising from their associated price movements within the sliding window. This structural difference means that, unlike NLP where context tokens vary, the financial model relies on the temporal/indicator dynamics of the same entities to generate representations.

CBOW Adaption Now CBOW is adapted as CBOS. This modification is applicable both temporally and across market/spatial axis.

C-CBOS Model In line with the models delineated previously, the model $F^{\langle \text{C-CBOS} \rangle}(.)$ is defined for estimating $x_i^{(t)}$ within a specified context $\{x_1^{(t)}, x_2^{(t)}, ..., x_{|C|-1}^{(t)}, x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}$.

The formal definition is given by

$$\begin{aligned} F^{\langle \text{C-CBOS} \rangle}&\left(\{x_1^{(t)}, x_2^{(t)}, \ldots, x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}\right) \\ &= \left(\sum_{x_j \in \{x_1^{(t)}, x_2^{(t)}, \ldots, x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}} (E[j] \star x_j) \right) \\ &\quad \cdot W_{\text{C-CBOS}}^T + \mathbf{b}_{\text{C-CBOS}} \end{aligned} \tag{6.15}$$

where $W_{\text{C-CBOS}} \in \mathbb{R}^{1 \times \xi}$ holds.

$\mathbb{F}$-C-CBOS To use $\mathbf{x}_i^{(t)}$ the model $F^{\langle \mathbb{F}\text{-C-CBOS} \rangle}$ is defined as

$$\begin{aligned} F^{\langle \mathbb{F}\text{-C-CBOS} \rangle}&(\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_{|C|}^{(t)}\} \setminus \{\mathbf{x}_i^{(t)}\}) \\ &= \begin{bmatrix} \sum_{x_j \in \{\mathbf{x}_1^{(t)}[1], \mathbf{x}_2^{(t)}[1], \ldots, \mathbf{x}_{|C|}^{(t)}[1]\} \setminus \{\mathbf{x}_i^{(t)}[1]\}} (E[j] \star x_j) \\ \vdots \\ \sum_{x_j \in \{\mathbf{x}_1^{(t)}[\mathbb{F}], \mathbf{x}_2^{(t)}[\mathbb{F}], \ldots, \mathbf{x}_{|C|}^{(t)}[\mathbb{F}]\} \setminus \{\mathbf{x}_i^{(t)}[\mathbb{F}]\}} (E[j] \star x_j) \end{bmatrix} \\ &\quad \cdot W_{\mathbb{F}\text{-C-CBOS}}^T + \mathbf{b}_{\mathbb{F}\text{-C-CBOS}} \end{aligned} \tag{6.16}$$

with $W_{\mathbb{F}\text{-C-CBOS}} \in \mathbb{R}^{\mathbb{F} \times (\mathbb{F} \cdot \xi)}$.

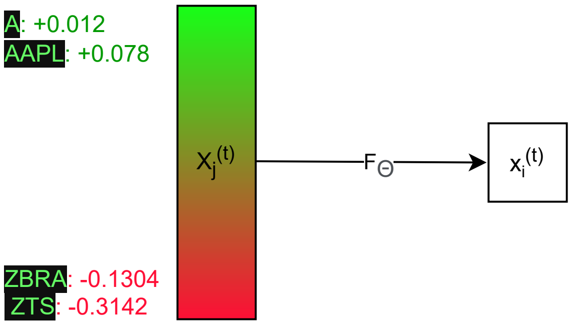


Figure 6.3: Schematic representation of the C-CBOS approach.

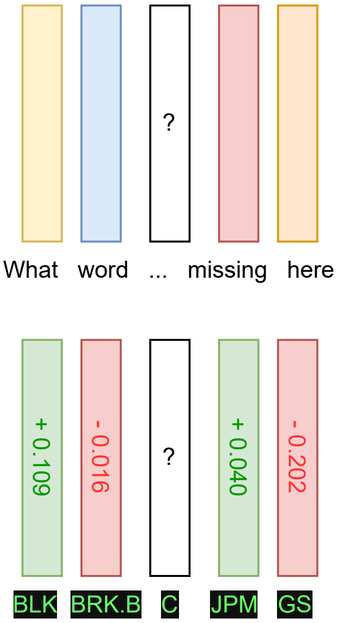


Figure 6.4: Schematic representation of the C-CBOS approach compared to an NLP approach.

As illustrated in Figure 6.3, the C-CBOS models aim to map the current market conditions as spatial information to the corresponding state of one stock. Similarly, as depicted in Figure 6.4, this approach aligns conceptually with the CBOW model, wherein each word is analogous to the current state of a stock. Again unlike words in the CBOW model, the stocks i.e. the position of each stock and the set of stocks in the C-CBOS framework remain consistent throughout the mapping process.

X-CBOS The $F^{\langle\text{X-CBOS}\rangle}(.)$ approach for predicting $x_i^{(t)}$ in a univariate time series requires a modification compared with the market axis. A temporal context integration into $\{x_i^{(t-\varpi)}, ..., x_i^{(t-1)}, x_i^{(t+1)}, ..., x_i^{(t+\varpi)}\}$ is needed; absent this, the methodology would reduce to a mere summation of scalars or OHCLV feature vectors, which is similar to a bag of features approach as outlined in [168] for dealing with non-stationarity. This has proven to be inferior in the experiments in this thesis. Without a temporal embedding, the model would collapse the sequence into an unordered sum of returns, losing information about the order of time steps. Such a setup would make the prediction task difficult, as the model would be unable to tell whether a particular input came from the recent past or from an earlier point within the context window. By introducing E_T, each observation is assigned a learnable temporal representation, ensuring that sequential order is preserved and that the model can use position-dependent dynamics rather than relying on undifferentiated scalar aggregations. Preliminary (untabulated) experiments confirmed this: when temporal embeddings were omitted, the models failed to learn because the input reduced to an unordered scalar sum without sequential information.

In NLP, however, the challenges are different. In the CBOW model, word embeddings are aggregated, yet the differentiation between these embeddings is sufficiently large as the index values of the word tokens and the embeddings differentiate from each other. This idea of distinctiveness also plays an important role in the X-CBOS approach in SF when using E_T which helps the model recognize and use the unique features of each time step.

Initially, a trainable time-step embedding matrix $E_T \in \mathbb{R}^{(\varpi\cdot 2)\times\xi}$ is introduced. Subsequently, the definition of the X-CBOS model is adjusted as

$$\begin{aligned} &F^{\langle\text{X-CBOS}\rangle}\left(\{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\} \setminus \{x_i^{(t)}\}\right) \\ &\qquad = \left(\sum_{x_i^{(j)} \in \{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\} \setminus \{x_i^{(t)}\}} (E[i] \star (E_T[j] \cdot x_i^{(j)}))\right) \\ &\qquad\qquad \cdot W_{\text{X-CBOS}}^T + \mathbf{b}_{\text{X-CBOS}} \end{aligned} \tag{6.17}$$

with $W_{\text{X-CBOS}} \in \mathbb{R}^{1\times\xi}$. This formulation allows the model to sum learned temporal representations, each scaled by the corresponding stock price, rather than aggregating scalar values directly.

$\mathbb{F}$-X-CBOS-Features The X-CBOS model taking $\mathbf{x}$ as an input is defined utilizing the same methodology as previously employed. Initially, a trainable time-step embedding matrix is introduced as $E_T \in \mathbb{R}^{(\varpi\cdot 2)\times\mathbb{F}\times\xi}$. This step ensures that, as in the other models, temporal order is preserved and jointly learned, rather than reducing the computation to a simple summation scalar values used for the scaling. The model is subsequently adjusted as

$$\begin{gathered} F^{\langle\mathbb{F}\text{-X-CBOS}\rangle}(\{\mathbf{x}_i^{(t-\varpi)},...,\mathbf{x}_i^{(t+\varpi)}\} \setminus \{\mathbf{x}_i^{(t)}\}) = \\ \begin{bmatrix} \sum_{x^{(j)}\in\{\mathbf{x}_i^{(t-\varpi)}[1],...,\mathbf{x}_i^{(t+\varpi)}[1]\}\setminus\{\mathbf{x}_i^{(t)}[1]\}}(E[i] \star (E_T[j,1]\cdot x^{(j)})) \\ \vdots \\ \sum_{x^{(j)}\in\{\mathbf{x}_i^{(t-\varpi)}[\mathbb{F}],...,\mathbf{x}_i^{(t+\varpi)}[\mathbb{F}]\}\setminus\{\mathbf{x}_i^{(t)}[\mathbb{F}]\}}(E[i] \star (E_T[j,\mathbb{F}]\cdot x^{(j)})) \end{bmatrix} \\ \cdot W^T_{\mathbb{F}\text{-CBOS}} + \mathbf{b}_{\mathbb{F}\text{-CBOS}} \end{gathered} \tag{6.18}$$

with $W_{\mathbb{F}\text{-CBOS}} \in \mathbb{R}^{\mathbb{F}\times(\mathbb{F}\cdot\xi)}$.

In the framework of the S2V X-CBOS model, both the $F^{\langle\text{X-CBOS}\rangle}$ and $F^{\langle\text{C-CBOS}\rangle}$ modules are employed to generate predictions for $x_i^{(t)}$. To synthesize the individual predictions, an averaging method is utilized, expressed as

$$F^{\langle\text{S2V-CBOS}\rangle}(x) = \frac{1}{2}F^{\langle\text{X-CBOS}\rangle}(.) + \frac{1}{2}F^{\langle\text{C-CBOS}\rangle}(.) \tag{6.19}$$

where $\mathcal{L}_{\text{S2V-CBOS}}$ denotes the combined predictive output of the model.

Similarly, this methodology is applicable to the feature-level predictions provided by $F^{\langle\mathbb{F}\text{-C-CBOS}\rangle}$ and $F^{\langle\mathbb{F}\text{-X-CBOS}\rangle}$.

Combining CBOS and SG In the domain of NLP, a methodology that can be adapted is the concurrent training of both SG and CBOW models. This approach necessitates the calculation of both tasks utilizing a shared embedding matrix E. The resultant loss functions, $\mathcal{L}_{\text{S2V-SG}}$ for SG and $\mathcal{L}_{\text{S2V-CBOS}}$ for CBOS, are integrated using a weighted sum to formulate the composite loss function, expressed as:

$$F_{\text{S2V}}(x) = \lambda_{\text{CBOS}} \cdot \mathcal{L}_{\text{S2V-CBOS}} + \lambda_{\text{SG}} \cdot \mathcal{L}_{\text{S2V-SG}} \,. \tag{6.20}$$

SMC Models S2V models can also be applied to SMC tasks, serving as a framework for both output predictions $\hat{\mathbf{y}}$ and input processing. Recent studies have demonstrated the feasibility of integrating categorical labels in lieu of continuous regression values within these models, as evidenced by implementations in high-frequency cryptocurrency analysis and time-series forecasting [180] [113]. Furthermore, the study by [257] introduces a co-movement prediction task, which bears resemblance to the SMC tasks discussed herein. It has been noted in [257] that the evolution of time series can be conceptualized as a stochastic process, resembling a random walk with tridirectional movement. This characterization aligns closely with methodologies in NLP, as indices replace the stock regression data, enhancing the model's parallels to NLP tasks. To incorporate SMC labels as input data, each time series element is transformed using $x_i^{(t)} \leftarrow \text{sign}(x_i^{(t)} - x_i^{(t+\omega)})$ with $\omega = 1$. Conceptually, this method parallels the NLP W2V framework, treating stock movements as binary outcomes with $V = \{0, 1\}$. The 'sentences' in this analogy are constructed either from $\left(\tilde{v}^{(i)}\right)_{i=1}^{|C|} \equiv X^{(t)}$ or $\left(\tilde{v}^{(i)}\right)_{i=1}^{\mathbb{T}} \equiv X_i$. If the SMC labels are used not only as targets but also as input scaling, the task of the model moves away from relating regression values to recognizing correlations of which stocks are likely to rise or fall together, as was done in [50], for example.

The determination of whether a stock price has risen or fallen depends on the specific model employed. The architecture of these models remain consistent, requiring only modifications to the architecture and the predictive targets to adapt to various SMC applications.

SG-SMC The $F^{\langle\text{C-SG-SMC}\rangle}$ is the SMC adapted variant of the $F^{\langle\text{C-SG}\rangle}(.)$ model and is defined as

$$F^{\langle\text{C-SG-SMC}\rangle}(x_i^{(t)}) = \sigma((E[i] \star x_i^{(t)}) \cdot W_{\text{C-SG-SMC}}^T + \mathbf{b}_{\text{C-SG-SMC}}) \tag{6.21}$$

, with $W_{\text{C-SG-SMC}} \in \mathbb{R}^{(|C|-1)\times\xi}$.

This implies that the architectural framework, along with all subsequent modifications, remains largely consistent with the original design. For SMC $\mathbf{y} = \mathbb{I}^{(t)}(X^{(t)} > X^{(t+\omega)})$ is set and $\mathcal{L}_{\text{C-SG-SMC}}(\hat{y}, y) = \mathcal{H}(\hat{\mathbf{y}}, \mathbf{y})$ is defined.

Next, the $F^{\langle\text{X-SG}\rangle}(.)$ is adapted as $F^{\langle\text{X-SG-SMC}\rangle}$ for SMC as

$$F^{\langle\text{X-SG-SMC}\rangle}(x_i^{(t)}) = \sigma((E[i] \star x_i^{(t)}) \cdot W^T_{\text{X-SG-SMC}} + \mathbf{b}_{\text{X-SG-SMC}}) \tag{6.22}$$

with $W_{\text{X-SG}} \in \mathbb{R}^{(2\cdot\varpi)\times\xi}$.

The target of the prediction is defined as

$$\mathbf{y}_{\text{X-SG-SMC}}[j] = \mathbb{I}^{(t)}(x_i^{(t)} > x_i^{(t-(\varpi-(j-1)))}) \tag{6.23}$$

with $1 \leq j \leq \varpi \cdot 2$ and for $\mathcal{L}_{\text{X-SG-SMC}}$ again $\mathcal{H}(.,.)$ is used.

CBOS-SMC The $F^{\langle\text{C-CBOS-SMC}\rangle}$ and $F^{\langle\text{X-CBOS-SMC}\rangle}$ models are defined as

$$\begin{aligned} F^{\langle\text{C-CBOS-SMC}\rangle}&\left(\{x_1^{(t)}, x_2^{(t)}, ..., x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}\right) \\ &= \sigma\left(\left(\sum_{x_j \in \{x_1^{(t)}, x_2^{(t)}, ..., x_{|C|}^{(t)}\}\setminus\{x_i^{(t)}\}} (E[j] \star x_j)\right)\right. \\ &\qquad \left. \cdot W^T_{\text{C-CBOS-SMC}} + \mathbf{b}_{\text{C-CBOS-SMC}}\right) \end{aligned} \tag{6.24}$$

with $W_{\text{C-CBOS-SMC}} \in \mathbb{R}^{1\times\xi}$ and

$$\begin{aligned} F^{\langle\text{X-CBOS-SMC}\rangle}&(\{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\} \setminus \{x_i^{(t)}\}) \\ &= \sigma\left(\left(\sum_{x_j \in \{x_i^{(t-\varpi)}, ..., x_i^{(t+\varpi)}\}\setminus\{x_i^{(t)}\}} (E[i] \star (E_T[j] \cdot x_i^j))\right)\right. \\ &\qquad \left. \cdot W^T_{\text{X-CBOS-SMC}} + \mathbf{b}_{\text{X-CBOS-SMC}}\right) \end{aligned} \tag{6.25}$$

with $W_{\text{X-CBOS-SMC}} \in \mathbb{R}^{1\times\xi}$.

Both receive

$$\mathbf{y}_{\text{CBOS-SMC}} = \mathbb{I}^{(t)}(x_i^{(t)} > x_i^{(t+1)}) \tag{6.26}$$

as a target and $\mathcal{H}(.,.)$ is used to calculate $\mathcal{L}_{\text{CBOS-SMC}}$.

All of the above models, predictions and loss functions can be combined as in the models before. The models are adapted for multiple features following the examples for the regressive tasks.

Vocabulary Based Approach The C-CBOS-SMC methodology is adjusted to better align with the NLP W2V frameworks. The target variable is expanded from a binary outcome to the full vocabulary. The target $\mathbf{y} \in \{0,1\}^{2\cdot|C|}$ is one hot encoded with $\mathbf{y} = 1$ for $j = \mathbb{I}^{(t)}(x_i^{(t)} > x_i^{(t+1)}) \cdot |C| + i$ with i being the currently sampled stock in the training step. $W_{\text{C-CBOS-SMC}}$ is extended to $W_{\text{C-CBOS-SMC-Vocab}} \in \mathbb{R}^{2\cdot|C|\times\xi}$ and the model is defined as

$$\begin{aligned} &F^{\langle\text{C-CBOS-SMC-Vocab}\rangle}\left(\{x_1^{(t)}, x_2^{(t)}, ..., x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}\right) \\ &\qquad = f_{\text{softmax}}\left(\left(\sum_{x_j \in \{x_1^{(t)}, x_2^{(t)}, ..., x_{|C|}^{(t)}\} \setminus \{x_i^{(t)}\}} \sigma(E[j] \star x_j)\right)\right. \\ &\qquad\qquad \left. \cdot\, W^T_{\text{C-CBOS-SMC}} + \mathbf{b}_{\text{C-CBOS-SMC}}\right) \end{aligned} \tag{6.27}$$

with $\mathcal{L}_{\text{C-CBOS-SMC-Vocab}}$ using the Cross-Entropy loss function.

For the $F^{\langle\mathbb{F}-\text{C-CBOS-SMC-Vocab}\rangle}$ model, an additional layer is implemented to enhance the integration of price feature information, as detailed in Section 6.11.2. This enhancement involves a learnable price feature embedding layer for each stock c_i which maps $\mathbf{x}_j^{(t)}$ to a representation which can be integrated to $E[j]$ via the $\star$-operation. Within this model, the target vector is defined as $\mathbf{y} \in \{0,1\}^{\mathbb{F}\times 2\cdot|C|}$ facilitating feature-specific predictions. Each feature within the model is predicted independently. The main idea of this approach is to build a target vocabulary that requires the model to identify both the correct company and its stock movement. This dual requirement is intended to align with NLP models and to address a limitation of the CBOS methodology.

In particular, this strategy is designed to circumvent the issue where the aggregation of $\mathbf{e}$-s in the CBOS approach leads to a market-centric representation rather than capturing distinctions specific to individual stocks per $\mathbf{e}_i$. The output vocabulary is conceptually defined as

$$V = \bigcup_{y=0}^{1} \{(c_i, y)\}_{i=0}^{|C|} \tag{6.28}$$

where the model has to choose the correct 'word' at each timestep. A schematic illustration of the approach can be seen in Figure 6.5.

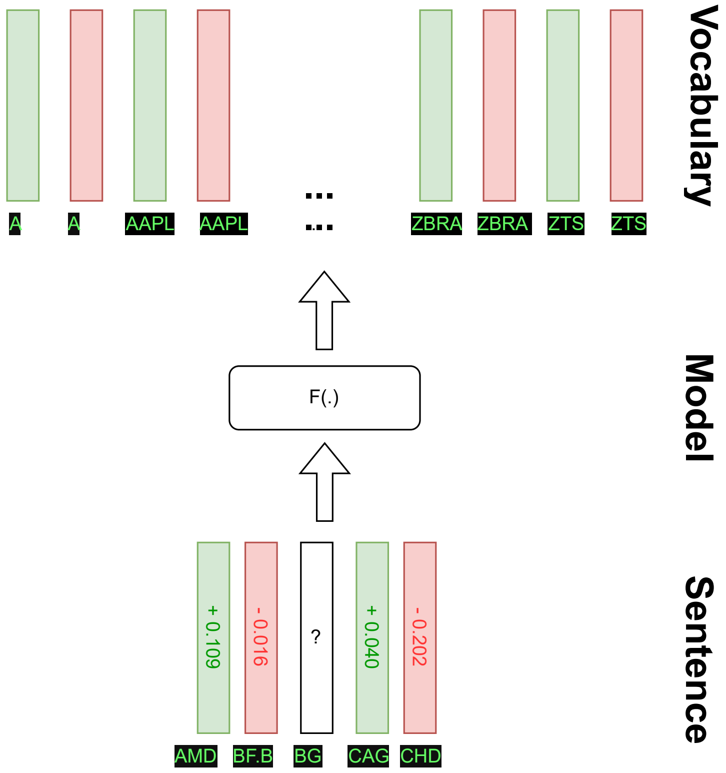


Figure 6.5: Schematic representation of the vocabulary-based approach.

6.7 Proposed Doc2Vec Adaption

This section is mainly based on the authors publication [223].

Doc2Vec models are designed to represent long sequences of word tokens $\tilde{w}^{(1)}, \ldots, \tilde{w}^{(\tilde{l})}$—from sentences to entire documents—as single high-dimensional dense vectors. This approach draws a parallel to W2V models, which similarly transform discrete textual elements into continuous vector spaces. For multivariate time series, aggregating data over Δt timesteps is interpreted analogously to Doc2Vec's document representation.

By analogizing documents to a finite collection of textual data $\{\tilde{w}^{(i)}\}_{i=1}^{\tilde{l}<\infty}$, the author defines 'market situations' as a fixed temporal segment of size Δt, consisting of sequential market snapshots. This adaptation was introduced as QMSEs in [223], with the aim of summarizing market dynamics in embeddings suitable for downstream ML models.

6.7.1 QMSE as Autoencoder Model

In [223] three different models are proposed to obtain a dense vector representation $\mathbf{e} \in \mathbb{R}^{\xi}$ of X, which are mainly based on encoder-decoder structures and will be briefly explained again below.

Autoencoder Model Doc2Vec models in NLP can generally be divided into two primary types. The first type includes less common models like Skip-Thought [109], which apply W2V techniques at the document level. The second type consists of encoder-decoder architectures, which are more commonly used in practice. These were already mentioned in [245] in the context of SF for the development of a distribution-based probabilistic modeling for latent variables. Due to the lack of successful outcomes from preliminary (untabulated) experiments with the first category, the focus has shifted towards encoder-decoder architectures. The encoder $F^{\langle \mathrm{E} \rangle}$ maps the input to a latent vector $\mathbf{e}$ intended to capture the main characteristics. This latent representation is then used by the decoder function $F^{\langle \mathrm{D} \rangle}$ to reconstruct the input as $\hat{\mathbf{y}}$, with $\mathbf{y} \equiv X$. For tasks involving textual data and stock price prediction, the use of recurrent architectures for both $F^{\langle \mathrm{E} \rangle}$ and $F^{\langle \mathrm{D} \rangle}$ appears intuitive. A key advantage of recurrent architectures lies in their capability to process input sequences of arbitrary length during both encoding and decoding phases. However, [223] shows that while recurrent encoder–decoders reconstruct well, they yield weaker representations $\mathbf{e}$; therefore, they are not pursued further here.

Established NLP Doc2Vec models can be utilized by treating $\tilde{X}$ directly as the embedding and circumventing the standard embedding process for $\tilde{w}^{(i)}$. This method parallels the embedding-based strategy applied to ASMs, as described in Section 6.11.1. While this direction was explored in a series of preliminary (and untabulated) experiments using Sentence-BERT [194], further investigation was ultimately decided against. The decision was based on the same shortcomings observed for RNN- and transformer-based models, namely unstable training dynamics, low pairwise distances between embeddings, the lack of coherent clustering structures, and the tendency to prioritize reconstruction accuracy over the representation of complex market behavior, as detailed in Section 7.4. The same holds true for Skip-Thought [109] adaptions. AEs, which are widely used for embedding training, excel at compressing high-dimensional information into more compact forms. For example, in the financial domain, Bao, Yue, and Rao [5] proposed an architecture based on LSTM networks that employs AEs to derive abstract and generalized representations of X.

The proposed AE model, denoted as $F^{\langle \mathrm{A} \rangle}$ is constructed by combining an encoder $F^{\langle \mathrm{E} \rangle}$ and a decoder $F^{\langle \mathrm{D} \rangle}$ and it is formally defined as

$$F^{\langle \mathrm{A} \rangle}(X) = F^{\langle \mathrm{D} \rangle}(F^{\langle \mathrm{E} \rangle}(f(X))) \tag{6.29}$$

where $F^{\langle \mathrm{D} \rangle}$ and $F^{\langle \mathrm{E} \rangle}$ are multilayer neural networks that utilize the tanh(.) activation function across ρ layers represented as $F_n^{\langle \mathrm{L} \rangle}$. The function $f(.)$ is employed to flatten the input, which has dimensions of either $|C| \times \Delta t$ or $|C| \times \Delta t \times \mathbb{F}$. Additionally, it holds that $\forall n : \dim(W_{F_n^{\langle \mathrm{L} \rangle}{}_D}) = \dim(W_{F_{\rho-n}^{\langle \mathrm{L} \rangle}{}_E})$, with $\dim(W_{F_n^{\langle \mathrm{L} \rangle}{}_D})[0] < |C| \cdot \Delta t(\cdot \mathbb{F})$.

To commence model training, the loss function is defined as $\mathcal{L}_A =$

$f_{\text{MSE}}(F^{\langle \text{A} \rangle}(X), X)$. In the present work, the model originally proposed in [223] is extended by incorporating SMC and utilizing RLRs. This modification is formally represented as

$$\mathcal{L}_A = \mathcal{H}(F^{\langle \text{A} \rangle}(\sigma(X)), \mathbb{I}(X \leq 0)) \text{ (using relative returns in the notation).} \quad (6.30)$$

This shifts the objective toward approximating SMC labels rather than reconstructing returns, aligning better with SMP. Furthermore, the QMSE $\mathbf{e} \in \mathbb{R}^{\xi}$ is expressed as $\mathbf{e} = F^{\langle \text{E} \rangle}(f(X))$.

6.7.2 QMSE Adaption

For the application of the QMSEs, scenarios have already been presented in [223], which are briefly summarized in the following.

CLM Adaption Within the domain of NLP, the utilization of CLM is often preferred over ULM, as opposed to the primary focus set in [220] which draws parallels between ULM and SF. CLM incorporates a distinct context Π to express $\tilde{w}^{(\tilde{l}+1)}$ and is defined by $\mathcal{P}_\theta(\mathcal{X} = \tilde{w}^{(\tilde{l}+\omega)} | \Pi, \tilde{w}^{(\tilde{l})}, \dots, \tilde{w}^{(1)})$ [75]. In the field of finance, particularly in TST/TSG analysis, the definition can be redefined as $\mathcal{P}_\theta(\mathcal{X} = X^{(t+1)} | \Pi, X^{(t)}, \dots, X^{(t-\Delta t)})$, with Π including fundamental data from previously mentioned sources. Research within financial analytics, such as [247] or the publications mentioned in Chapter 2, highlight the critical role of integrating such contextual information.

Similar to how document types are integrated into NLP models (as Π/through their representations in a Doc2Vec format), the present market scenario, when quantified as QMSEs for SF, can also be incorporated into the model. It is proposed that the methodology of representing ‘Situations’ as distributed embeddings aligns with the approach of event embedding models, as detailed in Section 7.4. In these models, structured event representations are extracted from essential news data, which enhance SMP processes. These event embeddings are broadly used as embedded representations of market conditions, thus forming a crucial complement to the quantitative approaches. Similarly to the CLM, where document type information might be encoded into the model as Π, the proposed QMSEs can be used as the context to SF models ($\Pi = \mathbf{e}$). As previously discussed in Section 3.0.5, incorporating global information into the model is vital for SF and is frequently employed to provide global information on the current market.

Learning Regularization In [223], methods to utilize QMSEs not directly for forecasting but as a technique to regularize the learning phase have been outlined. By recognizing unique scenarios, the model can tailor its learning strategy, given that current data might not accurately reflect 'traditional' stock market behaviors. This strategy aims to evaluate if the present training data, which could stem from unusual events, might prove ineffective for application and generalization across different scenarios.

Given non-stationarity and theories like RWT and EMH, it is reasonable to question if all market states should be labeled as exceptional. Arguments for the regularization approach are given in prior literature, such as [201], which explicitly acknowledges the existence of 'standard market conditions' [201] as a distinguishable baseline.

To determine whether the current training data stems from an unusual situation, its deviation from other QMSEs is analyzed. Experimental investigations suggest that calculating the distance from the entire set of QMSEs is not useful. Instead, the distance d to the previous κ QMSEs is computed. This method helps identify major deviations from normal market conditions, which may indicate an exceptional situation.

The calculation of d focuses on the distances among the embeddings $\mathbf{e}$ rather than the values of X, aiming to uncover underlying relationships and particularly movements that might not be detected by traditional metrics such as EMA or volatility indices. The regulatory model is formulated as

$$F^{\langle \mathrm{R} \rangle}(X^{(t)}) = \left\| \left(\frac{1}{\kappa} \cdot \sum_{j=t-(1+\kappa)}^{t-1} F^{\langle \mathrm{E} \rangle}(f(X^{(j)})) \right) - F^{\langle \mathrm{E} \rangle}(f(X^{(t)})) \right\|_2 . \tag{6.31}$$

With each successive t, the model integrates $X^{(t)}$ into the sliding window, continuously assessing its alignment with previously observed data. The value of d is directly linked to the extent of deviation from all other $\mathbf{e}$ vectors corresponding to the past κ market states. An increase in d thus reflects an exceptionally uncommon market condition relative to the current temporal context.

To incorporate this principle into the training process, the moderation of weight updates during backpropagation for atypical market scenarios is proposed. The rationale for moderating weight updates during backpropagation in atypical market scenarios lies in the assumed lower reproducibility of these situations. Market states that deviate strongly from recent conditions are assumed to be exceptional situations rather than structural patterns that generalize across time [223]. If such scenarios dominate the gradient signal during training, the model risks overfitting to these outliers, thereby lowering its ability to capture more stable dynamics. By scaling the loss function according to the deviation measure d, the proposed regularization reduces the influence of atypical cases while still allowing the model to incorporate them. This ensures that the learning process is not disproportionately driven by rare events, but rather maintains a balanced representation of both common and exceptional market states. Conceptually, this parallels robust training strategies in ML where sample reweighting or curriculum learning is employed to stabilize optimization under heterogeneous data distributions.

As outlined in [223], alternative strategies such as integrating QMSEs as additional context vectors or shifting embeddings within the prediction models did not yield improvements. Modulating weight updates during backpropagation, by contrast, provides a direct and architecture-independent mechanism to control the influence of atypical scenarios. Since the adjustment is derived directly from the embedding space, it leverages the same representational structure used for all market states, ensuring that the detection of atypicality and its impact on learning remain consistent and data-driven. This allows the model to adjust carefully when faced with unusual data, while keeping the same architecture and training loop. A pragmatic means of achieving this involves the proposed modification of the loss function as $\mathcal{L}_{\text{q-reg}} \leftarrow \mathcal{L} \cdot (1 + (1 + d)^{-1})$.

6.7.3 Further Experiments

Further potential applications of the QMSEs, previously explored in [223], are outlined. However, these approaches have been excluded from further consideration due to their limited relevance to SMP/SPP. The direct application of QMSEs through a Nearest Neighbor Similarity strategy (hereafter referred to as the Nearest Neighbors Approach, NNA) for SMP is briefly acknowledged but not elaborated on in this thesis due to its suboptimal results. As mentioned in Section 7.2 the evaluation of contextual embeddings presents a notable challenge, reflecting ongoing efforts within the research community to define quality metrics for word embeddings. One possible methodology involves examining nearest neighbors and determining whether the characteristics of these neighbors correspond to those of the original data point, thereby assessing whether the embedding effectively captures meaningful relationships.

The analysis is further extended to test the hypothesis that similar market situations are similar in the vector space, assuming that future stock movements can be inferred from current market conditions. As the results in [223] show, this assumption is probably not correct or only holds for a few cases. For the NNA the SMC labels are $\{-1, 1\}$.

Under the assumption that a small $\| \mathbf{e}^{(i)} - \mathbf{e}^{(j)} \|_2$ indicates similarity, $X^{(i)}$ and $X^{(j)}$ are considered similar, than $\mathbb{I}^{(i)}$ and $\mathbb{I}^{(j)}$ are expected to show similar behavior. Conversely, when $\| \mathbf{e}^{(i)} - \mathbf{e}^{(j)} \|_2$ is large, it suggests limited directional similarity, indicating a relationship where $\mathbb{I}^{(i)} \approx \mathbb{I}^{(j)} \cdot (-1)$ holds.

Based on these observations, the prediction of movements for $t + 1$ is defined as follows:

$$\hat{y} = \operatorname{sign}\left(\frac{1}{|K|} \cdot \sum_{\breve{d}_i \in K} (1 + \breve{d}_i)^{-1} \cdot \mathbb{I}^{(i)} \right) \tag{6.32}$$

where $K = \operatorname{topk}(\mathbb{E}, k)$ and $\mathbb{E} = \{\| \mathbf{e}^{(j)} - \mathbf{e}^{(t)} \|_2\}$. Alternatively, $\mathbb{E} = \{- \| \mathbf{e}^{(j)} - \mathbf{e}^{(t)} \|_2\}$ can be used to predict $\mathbb{I} \cdot (-1)$.

In another application scenario, there exists a potential for data reduction by encoding Δt with a subset of $\mathbf{e}$ vectors. These vectors constitute a dense, condensed, and unweighted aggregation of information, spanning extended temporal intervals, thereby encapsulating all essential information. The informational equivalence between $\mathbf{e}$ and X should be maintained, provided they correspond to identical temporal segments Δt. This would reduce input size and dimensionality, which is particularly interisting for resource-intensive transformers. This proposed conceptualization stands in contrast to the patching methodology employed by Zerveas et al. [263] and Nie et al. [165], as discussed in Section 3.0.4. Empirical tests in [223] showed degraded performance, so the strategy was not pursued further.

6.8 Proposed Recurrent Transformer

This section is mainly based on the authors publications [220] and [224].

Recent advancements in the field of ML, particularly in NLP, have been profoundly influenced by the development of transformer models and the attention mechanism. This architecture forms the basis for models such as GPT-3 [9], GPT-4 [173], LLaMA [216], and BERT [40], which have demonstrated remarkable capabilities. Given these developments, the transformer architecture is examined as an initial focus of this research.

As outlined in Section 6.3, a conceptual parallel is drawn between market snapshots and NLP tokens. This analogy aligns with the temporal orientation of the attention mechanism within transformer models, as highlighted by Li et al. [124]. This temporal aspect facilitates an enhanced understanding of sequential data, underscoring the transformative impact of attention mechanisms.

Zhao et al. [274] use a time-weighted function that gives greater importance to values close to the current or predicted data points. This method of time-weighting is implicitly integrated within all proposed transformer models that utilize the market axis for embedding representations. The attention mechanism within these models is exclusively oriented towards capturing temporal dependencies. This approach helps to 'discovered the non-linear relation between the importance and time point of stock data' [274].

To pivot the focus of the attention mechanism towards inter-stock dependencies, ASMs have been developed, which are detailed in Section 6.11. These models aim to incorporate relational dynamics between different stocks, thereby extending the utility of the attention mechanism beyond temporal analysis to include spatial stock interactions.

As delineated in Chapter 1, the utilization of recurrent transformer models to process extended temporal sequences is proposed in the research. The goal is to improve detection of non-stationary patterns in time series. Using longer intervals is expected to reveal dynamics that are otherwise treated as noise in short windows. It is also expected to expose long-term trends that are not visible in short sequences.

The processing of extended sequences in transformer architectures is computationally challenging, due to a time complexity of $\mathcal{O}(n^2 \cdot \xi)$ and a space complexity of $\mathcal{O}(n^2 + n \cdot \xi)$ of transformers [178]. This issue is notably pertinent in the domain of NLP, where the processing of lengthy inputs via transformers represents a significant area of ongoing research [8]. To address these computational constraints, the integration of recurrent transformers for SPP/SMP has been pioneered, as introduced in [224]. This adaptation iteratively processes long time series across multiple iterations κ.

The attention mechanism is implemented locally within discrete chunks, and a recurrent approach is employed across segments of the series to optimize computational resource utilization. To preserve and leverage information from preceding chunks, inspiration is drawn from RNNs and LSTM networks, incorporating a context Π in the model. This proposed context enables the storage and retrieval of information from previous iterations, enhancing the model's temporal coherence. The rationale for introducing Π is to enable the model to retain compressed representations of earlier chunks without directly carrying forward the raw data. In this way, Π functions as a recurrent state, analogous to the hidden or cell states in RNNs and LSTMs, and ensures that long-range dependencies are not lost when processing extended sequences in a chunk-wise fashion. This design allows the recurrent transformer to maintain temporal coherence across segments and to incorporate information from distant past observations in a computationally tractable manner. Conceptually, Π can be seen as a history embedding that parallels the role of context windows in CLM or MLM training (providing additional text as context), but specifically adapted to quantitative stock data. Abstracting prior information into Π is intended to capture slowly evolving dynamics (i.e. long-term dependencies) that are missed when attention is limited to a single chunk.

Incorporating distant time steps into the model through Π, rather than directly inputting the raw time series, can mitigate the challenges posed by non-stationarity. This approach minimizes potential numerical instabilities arising from the evolving characteristics of time series data, as the model processes the derived context rather than the time series itself. Nevertheless, the learning dynamics of the contextual representation remain susceptible to the underlying variability in the data.

In the MLM framework, as elucidated in Section 6.9, the task entails predicting a subsequent token, as $\mathcal{P}(\mathcal{X} = \tilde{w}^{(t)} | \tilde{w}^{(t-1)}, \ldots, \tilde{w}^{(t-\tilde{\varpi})}, \tilde{\Pi})$ with $\tilde{\Pi} = \{\tilde{w}^{(t+1)}, \ldots, \tilde{w}^{(t+\tilde{\varpi})}\}$. In the context of employing a recurrent transformer architecture, $\tilde{\Pi}$ is redefined to encompass $\tilde{\Pi} = \{\tilde{w}^{(t-\tilde{\varpi})}, \ldots, \tilde{w}^{(t-\tilde{\Delta}t)}\}$, thus incorporating tokens from a more extended historical context into the predictive model. This proposed adaptation contrasts markedly with an MLM adaption (using future stock price information), wherein access to context from the distant past is available within practical applications of the model. This design is intended to integrate long-range temporal dependencies and improve predictive accuracy. Chunk-wise processing is also expected to improve robustness to noise, consistent with [186], where permutation is used as an auxiliary task. The recurrent transformer, denoted as $F^{\langle \mathrm{T} \rangle}(.)$, incorporates ρ encoder layers $(F_1^{\langle \mathrm{E} \rangle}(.), \ldots, F_\rho^{\langle \mathrm{E} \rangle}(.))$ where each layer is defined according to the specifications in [219].

In the given model, Π, which encapsulates recurrence by incorporating input from the preceding iteration $\kappa - 1$, can be implemented in one of two distinct methodologies. One option is to supply the context once at the start, $F^{\langle \mathrm{T} \rangle}(\Pi^{(\kappa)}, \mathsf{X}^{(\kappa)})$, which sets a shared context for the sequence. Alternatively, the context can be iteratively outputted and utilized within each individual transformer encoder block, denoted as $F_n^{\langle \mathrm{E} \rangle}(\Pi_n^{(\kappa)}, Z_{n-1})$, where Z serves as a generalized placeholder representing the output from the previous encoder stage of the respective model. 'N' is used for models adopting this procedural approach.

The context is initialized with a uniform Xavier [77] distribution as

$$\Pi[i,j] \sim \mathcal{U}\left(-\sqrt{\frac{\theta}{n_{\text{in}}+n_{\text{out}}}}, \sqrt{\frac{\theta}{n_{\text{in}}+n_{\text{out}}}}\right)^{\xi \times \phi_{\Pi}} \quad \text{with } \theta = 6 \ . \tag{6.33}$$

The stock data input $\mathsf{X}^{(\kappa)}$ is defined as

$$\mathsf{X}^{(\kappa)}[i,j] = X[i,(\kappa-1)\cdot\theta_{\kappa}+j] \tag{6.34}$$

for each κ respectively, where θ_κ is the chunk size. The input is recursively fed into $F^{\langle \mathrm{T} \rangle}(.)$ for $\max(\kappa)$ iterations until the termination condition $\kappa > \frac{\Delta t}{\theta_\kappa}$ is satisfied. During each iteration, $F^{\langle \mathrm{T} \rangle}(.)$ receives $\Pi^{(\kappa)}$, from the preceding iteration.

It is indicated by the experimental results that the most effective normalization strategy for the chunk $\mathsf{X}^{(\kappa)}$ is the one proposed by [141]. This observation stands in contrast to the optimal normalization approach for ASMs which work best with the one from Appendix A.1.

Furthermore, the incorporation of $F^{\langle \mathrm{LL} \rangle}$ proves to be essential for processing $\mathsf{X}^{(\kappa)}$. Specifically, a latent transformation with a tanh(.) activation function, or the one described in Section 6.8, is employed in the authors implementation. Without this preprocessing step, the model exhibits significant difficulty in learning meaningful representations. The instability observed when omitting $F^{\langle \mathrm{LL} \rangle}$ is consistent with the findings of [253], which previously demonstrated that multi-view data can exhibit instability in subsequent modeling stages when not subjected to adequate preprocessing. Analogous to this in the NLP area, one would probably not create a language model without a (learnable) word embedding matrix.

Attention-Mechanism In modern NLP architectures, self-attention mechanisms conventionally employ multiple attention heads, each selectively attending to distinct subregions of the input sequence. This design promotes a decomposition of representations into multiple latent subspaces, resulting in a more detailed encoding. However, a fundamental deviation from conventional NLP methodologies is present in the proposed models due to the structural composition of its input representation. Specifically, whereas information in standard tokenized sequences is distributed across multiple discrete tokens, a stacked input paradigm is employed, wherein all spatial and indicator-related information corresponding to a single timestep is encapsulated within a singular token.

Consequently, this structural distinction, combined with the empirical findings in this thesis (cf. Section 7.5.1) during pretraining, suggests that a single-head attention mechanism—similar to the approach proposed in [115]—is more appropriate in this context. This departs from findings such as [134] (Chapter 2), which report benefits of multi-head attention for stock data. An alternative developed approach proposed is to modify $F^{\langle \text{LL} \rangle}$ as $F^{\langle \text{L} \rangle}$ using the weight parameter $W_L \in \mathbb{R}^{\rho_{\text{heads}}+1\times\lfloor\frac{\xi}{\rho_{\text{heads}}}\rfloor\times\xi}$ (assuming $\eta = |C| \cdot \mathbb{F}$ holds). This is used to calculate

$$\dot{X}_h = X^T \cdot (W_L[h])^T \tag{6.35}$$

and

$$\bar{X}[j,t] = \tanh((\dot{X}_h)^T[i,t]) \tag{6.36}$$

with $j = (h-1) + i$[1].

This method ensures that the semantic information embedded within a market snapshot is redundantly encoded in two ways: First, as in conventional multi-head attention, the same input is projected into multiple distinct subspaces—one per attention head—where each head processes the information independently. Second, prior to the attention operation itself, a head-specific latent transformation is applied. Together, these two stages ensure that each market snapshot is represented multiple times across both latent and attention layers.

The context Π with length ϕ_Π may be integrated in one of two fashions: it can either be concatenated, in what is referred to as merged-attention (abbreviated with 'M'), with the input, or employed as queries in cross attention (abbreviated with 'C') mechanisms within $F_n^{\langle \text{E} \rangle}$. In the scenario utilizing cross attention, it is infeasible to establish a distinct context that can be updated in relation to the data from the preceding iteration κ.

[1] For the ρ_{heads}-th head, not all latent representations may be present to fit with the dimension number.

For the merged attention based models the inputs are fed in the attention-head $\text{head}_{\rho_{\text{head}}}$ in the layer $F_n^{\langle\text{E-M}\rangle}(.)$ (as defined in [219] with slight modifications in the notation) as

$$F^{\langle\text{Attention}\rangle}\left(\left[\left(\Pi^{(\kappa)}\right)^T \odot \left(\mathsf{X}^{(\kappa)}\right)^T\right] \cdot W_Q, \left[\left(\Pi^{(\kappa)}\right)^T \odot \left(\mathsf{X}^{(\kappa)}\right)^T\right] \cdot W_K, \left[\left(\Pi^{(\kappa)}\right)^T \odot \left(\mathsf{X}^{(\kappa)}\right)^T\right] \cdot W_V\right) \tag{6.37}$$

where $\Pi^{(\kappa)}$ is defined according to the respective model version.

For cross attention based models the definition simplifies to

$$\text{head}_{\rho_{\text{head}}} = F^{\langle\text{Attention}\rangle}\left(\left(\Pi^{(\kappa)}\right)^T \cdot W^Q, \left(\Pi^{(\kappa)}\right)^T \cdot W^K, \left(\mathsf{X}^{(\kappa)}\right)^T \cdot W^V\right) \ . \tag{6.38}$$

The final output of both merged attention approaches and cross attention approaches is transposed again so $F_n^{\langle\text{E-M}\rangle}(Z) \in \mathbb{R}^{\xi\times(\phi_\Pi+\theta_\kappa)}$ and consequently $F_n^{\langle\text{E-C}\rangle}(Z) \in \mathbb{R}^{\xi\times\theta_\kappa}$ holds true.

In examining the architectural variations of the proposed recurrent transformer models, the primary distinction (next to the cross attention, merged attention and multi context) lies in the mechanisms employed for context generation. Three distinct models from the domain of RNNs are proposed to illustrate these variations. The most rudimentary of these models derive from adaptations of the Jordan models [105], denoted hereafter with 'J'. These models exhibit conceptual similarities with the methodologies employed in TransformerXL [31]. Other models adapt techniques from Elman RNNs [58] (denoted 'R'). These bear resemblance to the Recurrent Memory Transformer model [10]. Additionally, models based on LSTMs [87], indicated by 'L', parallel the structure found in Block-Recurrent Transformers [94]. Position encodings listed in Section 6.4 are employed. This observation aligns with the reasoning presented in [169], which emphasizes the efficacy of such encodings in capturing structural dependencies within the data.

$\Pi^{(\kappa)}$ Definition The architectures presented in the following are illustrated in Figure 6.6; 1) are the models from Section 6.8, 2) those from section 6.8 and 3) those from section 6.8.

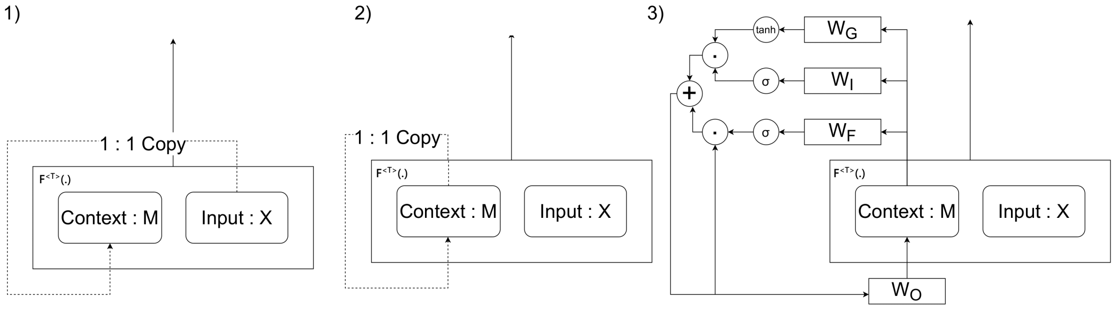


FIGURE 6.6: Schematic illustration of the recurrent transformer architectures. The illustration is taken from the authors publication [224].

Jordan based Π Assignment The 'J' in $F^{\langle \mathrm{J} \rangle}$ denotes the Jordan Network inspired models [105], which are precursors to RNNs. In these networks, the entirety of the output is fed back into the model as an exact copy during the subsequent iteration, concurrently with the input of the new iteration step.
For $F^{\langle \text{J-M} \rangle}$ models

$$\Pi^{(\kappa)}_{\text{J-M}}[i,j] = F^{\langle \text{J-M} \rangle}_{\rho}(\Pi^{(\kappa-1)}_{\text{J-M}}, Z)[i,j] \tag{6.39}$$

with $j \geq \theta_\kappa$ and $Z \in \mathbb{R}^{\xi \times \theta_\kappa}$ is defined. This extracts for $F^{\langle \text{J-M} \rangle}(.)$ the positions generated by $\mathsf{X}^{(\kappa)}$ as part of the output for Π in iteration κ+1. For $F^{\langle \mathrm{N} \rangle}$ approaches this translates to

$$\Pi^{(\kappa)}_{\text{J-M-N}}[n,i,j] = F^{\langle \text{J-M-N} \rangle}_{n}(\Pi^{(\kappa-1)}_{\text{J-M-N}}[n], Z)[i,j] \ . \tag{6.40}$$

In the context of these models, employing cross attention involves feeding the entire output of $F^{\langle \mathrm{E} \rangle}_{\rho}$ from iteration step κ into the system, as delineated in Section 6.8. This output is then utilized as the queries within the attention mechanism.
Therefore

$$\Pi^{(\kappa)}_{\text{J-C}} = F^{\langle \text{J-C} \rangle}_{\rho}(Z^{(\kappa-1)}) \tag{6.41}$$

holds true and for $F^{\langle \text{J-C-N} \rangle}$, $\Pi \in \mathbb{R}^{\rho \times \phi_\Pi \times \xi}$ is defined as

$$\Pi^{(\kappa)}[n]_{\text{J-C-N}} = F^{\langle \text{J-C-N} \rangle}_{n}(Z^{(\kappa-1)}) \quad . \tag{6.42}$$

Elman RNN based Π Assignment In the $F^{\langle \mathrm{R} \rangle}$-models the latent representation of the previous κ is fed in the model. This contrasts to the Jordan Networks, where the entirety of the output is reintroduced into the model, potentially leading to noise carousels and error feedback loops. Cross attention is not possible for these models. This limitation occurs because no contextual learning can take place if the model simply reproduces the modified input, preventing the use of cross-attention mechanisms.

The $F^{\langle\text{R-M}\rangle}$ models define $\Pi^{(\kappa)}$ as

$$\Pi^{(\kappa)}_{\text{R-M}}[i,j] = F^{\langle\text{R-M}\rangle}_{\rho}(\Pi^{(\kappa-1)}_{\text{R-M}}, Z)[i,j] \tag{6.43}$$

for $j \leq \phi_{\Pi}$. For $F^{\langle\text{R-M-N}\rangle}$ models

$$\Pi^{(\kappa)}_{\text{R-M-N}}[n,i,j] = F^{\langle\text{R-M-N}\rangle}_{n}(\Pi^{(\kappa-1)}_{\text{R-M-N}}[n], Z)[i,j] \tag{6.44}$$

(again with $j \leq \phi_{\Pi}$) is defined.

LSTM based Π Assignment The final model class draws on the LSTM architecture [86]. While it does not include all aspects of the traditional LSTM framework, it selectively incorporates key components that are useful for generating Π. Of particular interest is the adoption of the 'error carousel' mechanism, a pivotal feature of LSTM models that facilitates effective error propagation and learning stability.

$\Pi^{(\kappa)}$ is defined at iteration step κ as

$$\Pi^{(\kappa)} = \tanh(\ddot{\Pi} \cdot W_O^T + \mathbf{b}_O) \tag{6.45}$$

with $W_O \in \mathbb{R}^{\xi\times\xi}$.

This calculation is inspired by the 'Output-Gate' of LSTM models. Thus, Π takes the role of the long term memory or 'cell-state'. The intermediate result after processing the transformer encoder layer is defined as $\acute{\Pi} = F^{\langle\text{E-L}\rangle}_{\rho}(Z)$ to improve readability in the following.

To update this cell state $\acute{\Pi}^{(\kappa)}$ and the 'forget-gate' is used. This calculates as

$$\mathcal{F}^{(\kappa)} = \sigma(\acute{\Pi}^{(\kappa-1)} \cdot W_F^T + \mathbf{b}_F) \tag{6.46}$$

with $W_F \in \mathbb{R}^{\xi \times \xi}$.

Also inspired by the LSTM model the mechanisms

$$I^{(\kappa)} = \sigma(\acute{\Pi}^{(\kappa)} \cdot W_I^T + \mathbf{b}_I) \tag{6.47}$$

with $W_I \in \mathbb{R}^{\xi \times \xi}$ and

$$G^{(\kappa)} = \tanh(\acute{\Pi}^{(\kappa)} \cdot W_G^T + \mathbf{b}_G) \tag{6.48}$$

with $W_G \in \mathbb{R}^{\xi \times \xi}$ are used to represent the input gate. With all these auxiliary variables the update function for the cell state of the next iteration $\kappa + 1$ can be defined as

$$\ddot{\Pi}^{(\kappa)} = \mathcal{F}^{(\kappa)} \otimes \acute{\Pi}^{(\kappa)} + G^{(\kappa)} \otimes I^{(\kappa)}\ . \tag{6.49}$$

The proposed use of a forget gate, inspired by the LSTM model, allows the model to selectively remove parts of the previous cell state, making room for new contextual information as controlled by the input gate. This mechanism improves the model's ability to adapt to changing data patterns by balancing information retention and removal.

Non-recurrent Transformer / Baseline Transformer For the $F^{\langle \mathrm{M} \rangle}$ models, the parameters can be assigned such that $\theta_\kappa = \Delta t$ and $\phi_\Pi = 0$, thereby establishing a non-recurrent architecture (i.e. $F^{\langle \text{BM-T} \rangle}$). This configuration allows the impact of recurrence on performance to be evaluated. Comparing this non-recurrent model with its recurrent counterparts enables a quantitative assessment of recurrence.

6.9 Proposed Pretrained Transformer

This section is mainly based on the authors publications [220] and [224].

6.9.1 Masking Tasks

Arguably the most prevalent pretraining task for language models is MLM. Here, individual word tokens $I_{\text{MLM}} = \{i \in \mathbb{N}, i < \tilde{l}, M[i] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\text{MLM}})\}$ are masked and the new tokenized input text $(\tilde{\mathbf{e}}^{(i)})_{i=0}^{\tilde{l}} : \forall i \in I_{\text{MLM}} : \tilde{\mathbf{e}}^{(i)} \leftarrow \tilde{\mathbf{e}}_{[\text{MASK}]}$ is then processed through $\tilde{F}^{\langle\text{SM}\rangle}(.)$. Subsequently, all indices $i \in I_{\text{MLM}}$ are analyzed by an MLM head designed to predict the original word token at each masked position. Similar to the proposed S2V model, an adapted masking task can be applied along both the temporal and the spatial/market axes.

As outlined in Chapter 1, masking on the market axis is intended to help the model capture delayed cross-asset correlations by reconstructing masked prices from other c_i. Temporal masking and the inclusion of future prices as context are motivated by CLM, of which MLM is a variant. Here $\mathcal{P}(\mathcal{X} = \tilde{w}^{(t)} | \tilde{w}^{(t-1)}, \dots, \tilde{w}^{(t-\tilde{\varpi})}, \tilde{\Pi})$ with $\tilde{\Pi} = \{\tilde{w}^{(t+1)}, \dots, \tilde{w}^{(t+\tilde{\varpi})}\}$ is calculated.

The approach can be adapted by adding $\Pi = \{X^{(t+1)}, \dots, X^{(t+\varpi)}\}$ as an additional contextual element. Although future stock values cannot be obtained in real-world applications, this proposed modification is designed to enhance the model's capabilities in several areas. Firstly, by incorporating additional information, the model is expected to identify and represent market dynamics and patterns that might otherwise be treated as noise. Secondly, the inclusion of future values as a teaching mechanism introduces the model to concepts of randomness and stochastic processes, thus fostering an understanding that some elements remain unpredictable. Thirdly, the model is trained to recognize the market's behavior in response to unforeseen events. While it is not expected to predict these events, the model can generate informed predictions about subsequent market behavior and learn about the interdependencies among variables [263].

This approach also permits retrospective analysis of temporal dynamics that would otherwise remain unexplained. The insights derived from such retrospective analysis can subsequently be encoded into the model as abstract knowledge, potentially enhancing its capacity to predict and interpret future movements at other temporal points within the context of SMP/SPP. Transferred to the NLP area, one would not simply perform next token prediction (only SPP/SMP without knowledge of future stock trends) as a training task, but also MLM to generate generalized knowledge.

Furthermore, the application of masking facilitates an increase in the quantity of available training samples. This is achieved by passing the same $\grave{X}^{\xi\times\Delta t}$ multiple times through the model, each time representing a distinct data sample. Such an approach is highlighted in [249], where masking nodes serves as a foundational motivation for the implementation within their GNN-based model. As noted in Chapter 1, the strategy is particularly useful when only limited data are available (e.g. for interday). For all masking tasks, a mask $M \in \{0,1\}^{\dim(X)}$ is employed. The input to the model is defined as $\acute{M} \otimes X + \hat{M}$, with $\hat{M} = m \cdot M$ and $\acute{M} = 1 - M$, where m denotes a learnable embedding parameter representing masked values.

The overarching aim of all masking tasks is to facilitate the reconstruction of masked inputs. This objective is pursued through the optimization process defined by the minimization of the loss function $\mathcal{L}_{\mathrm{M}} = \frac{1}{\mathbf{1}^T M \mathbf{1}}(\sum_{i=1}^{\xi}\sum_{j=1}^{\Delta t}(F^{\langle \mathrm{T}\rangle}(X) \otimes M)[i,j] - (X \otimes M)[i,j])^2$ for SME (the BCE is used for SMC accordingly). For certain masking tasks, additional heads can be added; details are given where relevant. A suite of masking tasks is introduced in this research:

The task of Masked Feature Modeling (MFM) is defined as $\forall i,j : M[i,j] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\mathrm{MFM}})$ with $i \in \mathbb{N} \leq \xi, j \in \mathbb{N} \leq \Delta t$. In this setting, the model receives both past and future points for the target stock and the same features of other stocks within the sliding window. This dataset not only includes the target stock's price features but also extends to the analogous features of other stocks within the sliding window.

For MFM and the subsequently introduced Masked Price Modeling (MPM), a head is proposed to enhance the model's expressiveness for prediction. $H \in \mathbb{R}^{\dim(X)}$ is fed into an RNN followed by a linear layer, yielding $\dot{H} \in \mathbb{R}^{\dim(H)}$. A learnable parameter $W_M \in \mathbb{R}^{\xi \times \Delta t \times \Delta t}$ is introduced and used to compute

$$\ddot{H}[i] = \tanh(\dot{H}[i]) \cdot W_M[i])^T \ . \tag{6.50}$$

This allows each sliding window of each feature and each stock to be embedded separately by a dedicated head, rather than relying solely on the transformer's output. This is particularly beneficial since the proposed transformer's output—especially after passing through the ReLU and dropout layers—may struggle to accurately represent the results.

Masked Timestep Modeling (MTM) is defined (i.e. the cloze typ masking [263]) as $\forall i \in \mathbb{N} < \Delta t : \mathbf{b}[i] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\text{MTM}})$ with $\dim(\mathbf{b}) = \Delta t$ and $M[i,j] = \mathbf{b}[j]$.

For MTM, an optional linear layer is defined with an initial $\sigma(.)$ activation function processing batch wise inputs as $F^{\langle \text{MTM} \rangle}(.)$ having a weight $W_{\text{MTM}} \in \mathbb{R}^{\xi \times \xi}$. Here R_{MTM} is the input, which is defined as

$$I_{\text{MTM}} = \{j | \mathbf{b}[j] = 1\}, \quad \forall i \in I_{MTM} : R_{\text{MTM}}[k,i] = F^{\langle \text{T} \rangle}(X)[k,i] \tag{6.51}$$

and redefine

$$\forall i \in I_{MTM} : F^{\langle \text{T} \rangle}(X)[k,i] \leftarrow F^{\langle \text{MTM} \rangle}(R_{\text{MTM}})[k,i] \ . \tag{6.52}$$

Next, Masked Stock Modeling (MSM) is defined as $\forall i,j : B_{\text{MSM}}[i,j] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\text{MSM}})$ with $\dim(B_{\text{MSM}}) = (\xi, \max(\kappa))$ and

$$\forall i \in \mathbb{N} < \xi, \forall j \in \mathbb{N} < \max(\kappa), \forall k \in \mathbb{N} < \theta_\kappa : M[i, (j-1) \cdot \theta_\kappa + k] = B_{\text{MSM}}[i,j] \ . \tag{6.53}$$

Again a batch wise linear layer $F^{\langle \mathrm{MSM} \rangle}(R_{\mathrm{MSM}})$ is defined and this time

$$\begin{aligned} I_{\mathrm{MSM}} = \{(i,j) | B_{\mathrm{MSM}}[i,j] = 1\} \\ (i_k, j_k) \leftarrow \phi(k) \\ \forall k \in \mathbb{N} \leq |I_{\mathrm{MSM}}|, \forall w \in \mathbb{N} < \theta_\kappa : R_{\mathrm{MSM}}[k,w] = \\ F^{\langle \mathrm{T} \rangle}(X)[\phi(k)[1], (\phi(k)[2] - 1) \cdot \theta_\kappa + w] \end{aligned}$$

holds. The outputs of $F^{\langle \mathrm{MSM} \rangle}(R_{\mathrm{MSM}})$ are assigned in the models output as before. Conceptually, MSM can be compared to the masking of longer sentence fragments in NLP, as described in [106]. Due to the poor performance of stock masking, a head per stock is proposed, as in MPM/MFM, and the entire time series was embedded, but without success.

Further Masked Price Modeling (MPM) is defined as $\forall i,j : B_{\mathrm{MPM}}[i,j] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\mathrm{MPM}})$ with $\dim(B_{\mathrm{MPM}}) = (|C|, \Delta t)$ and

$$\forall f \in \mathbb{N} < \mathbb{F} + 1 : M\left[|C| \cdot (f-1) + i, j\right] = B_{\mathrm{MPM}}[i,j] \ . \tag{6.54}$$

Moreover Patch Masked Modeling (PMM) is defined as

$$L_{\mathrm{PMM}} \subseteq (\mathbb{N}_0 \cap [0, \xi)) \Join (\mathbb{N}_0 \cap [0, \Delta t)) \Join \{i \in \mathbb{R} | 0 \leq i \leq 1\} \tag{6.55}$$

and

$$\begin{aligned} \forall l_k \in L_{\mathrm{PMM}} : \\ B_{\mathrm{PMM},k}[i,j] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(l_k[3]) \text{ with } i \in \mathbb{N}_0 < \frac{\xi}{l_k[1]}, j \in \mathbb{N}_0 < \frac{\Delta t}{l_k[2]} \\ \forall w \in \mathbb{N} \leq l_k[1], \forall v \in \mathbb{N} \leq l_k[2] : M_k[i \cdot l_k[1] + w,\ j \cdot l_k[2] + v] = B_{\mathrm{PMM},k}[i,j] \end{aligned} \tag{6.56}$$

with

$$M = \mathrm{sign}(\sum_{k=1}^{|L_{\mathrm{PMM}}|} M_k) \ . \tag{6.57}$$

For l_k where $l_k[1] = l_k[2]$ holds, a linear layer $F^{\langle \mathrm{PMM} \rangle}(.)$ can be defined.

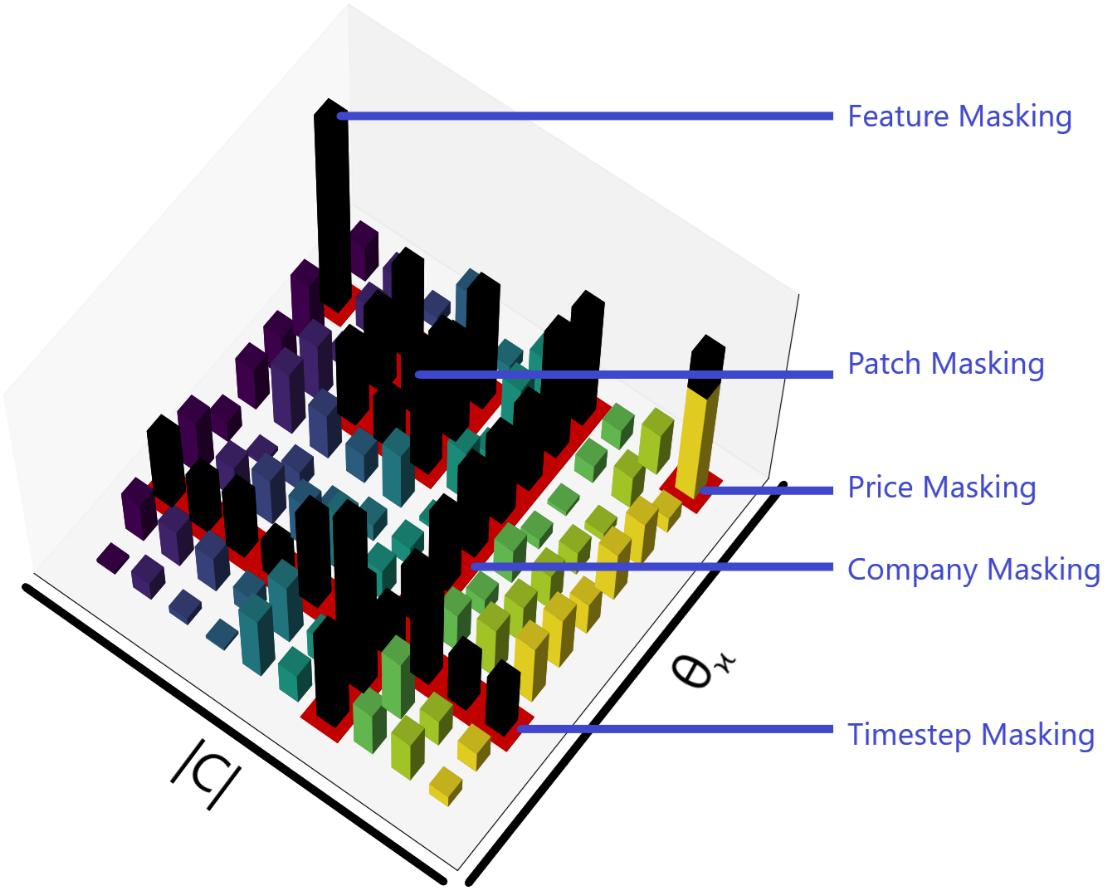


Figure 6.7: Conceptual mapping of the various masking tasks. One color represents the price performance of a stock over a period of time. Black implies the respective masking.

A graphical representation of all these methodologies is provided in Figure 6.7.

6.9.2 Trend-Matching

For training many language models in the NLP domain, the NSP task is used. Two sequences of word-tokens $\left(\tilde{w}^{(1)}, \tilde{w}^{(2)}, \ldots, \tilde{w}^{(\tilde{l})}\right)$ and $\left(\tilde{w}^{(\tilde{l}+1)}, \tilde{w}^{(\tilde{l}+2)}, \ldots, \tilde{w}^{(\tilde{l}+\tilde{l}_2)}\right)$ are entered into the model concatenated with a (separator) '[SEP]' token $F^{\langle \tilde{\mathrm{S}}\mathrm{M}\rangle}\left(\tilde{w}^1, \tilde{w}^2, \ldots, \tilde{w}^{\tilde{l}} \odot \tilde{h}_{[\mathrm{SEP}]} \odot \tilde{w}^{\tilde{l}+1}, \tilde{w}^{\tilde{l}+2}, \ldots, \tilde{w}^{\tilde{l}+\tilde{l}_2}\right)$ (following the notation of Section 6.1).

The position of the [SEP]-token is subsequently extracted from the computational output. This extracted position is then inputted into a NSP module. A binary decision is made by the NSP module on whether the two sequences are contiguous parts of the same text.

Formally

$$(F^{\langle \widetilde{\text{SM}} \rangle}[i, \tilde{l} + 1]) \cdot W_{\text{NSP}}^{T} + \mathbf{b}_{\text{NSP}} \tag{6.58}$$

with $i \leq \tilde{\xi}$ and $W_{\text{NSP}} \in \mathbb{R}^{\tilde{\xi} \times 2}$ is calculated.

The task can be adapted to the proposed transformer models to encourage learning of macro-level market regularities (analogous to NSP in NLP). As noted earlier in Chapter 1, the main goal of training this proposed TM task is to help the model develop an understanding of macroeconomic principles, similar to how NSP in NLP improves semantic understanding in language. The paradigm is intended to support longer-range forecasting i.e. $\omega > t + 1$. The approach is also intended to improve robustness to stochastic variation by emphasizing stable regularities over noise.

For the approach, a trend-matching token, denoted as $\mathbf{h}_{\text{TM}} \in \mathbb{R}^{\xi}$, is inserted within the current training mini-batch between two sections of X. At each timestep the model input is redefined as

$$X \leftarrow \left[\dot{X} \quad \odot \quad \mathbf{h}_{\text{TM}} \quad \odot \quad \ddot{X} \right] . \tag{6.59}$$

The two segments of the dataset are concatenated alongside the token, subsequently forming the input for the model $F^{\langle \text{T} \rangle}$. Following this integration, the position of the [TM] token is ascertained and subsequently inputted into a linear layer. This layer is responsible for executing a binary decision, determining whether the two stock trends under consideration sequentially follow each other. By using (relative) returns or RLR, the model cannot recognize jumps in the data and then make a 'Clever Hans' prediction [135], which would have been possible on the absolute stock prices due to sudden large differences between the time steps before and after $\mathbf{h}_{\text{TM}}$.

In the following let v be the timestep assigned for X wrt. to $\grave{X}$. Also the length of one segment is set to be $\theta_{\text{TM}} \approx \frac{1}{2} \cdot \Delta t$. Next the random predecessor trend start index is set to

$$u = b \cdot (\ \mathcal{U}((\mathbb{N} < (\mathbb{T} - \Delta t)) \setminus \{v\})) + (1 - b) \cdot (v + \theta_{\text{TM}}) \tag{6.60}$$

where $b \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_{\text{TM}})$ holds.

Further

$$\dot{X}[i,k] = \grave{X}[i, v+k] \ , \tag{6.61}$$

$$\ddot{X}[i,+k] = \grave{X}[i, u+k] \tag{6.62}$$

with $\forall k \in \mathbb{N}_0 < \theta_{\text{TM}}$ holds.

The underlying concept of the TM shares similarities with the approach in [169], albeit in the reverse direction. In that work, historical time events are analyzed for similarity, and those deemed dissimilar are suppressed in subsequent training. Additional examples of contrastive learning tasks can be found in [177].

6.9.3 Finetuning

Fine-tuning is treated as an SF task with $\omega = 1$ (SPP or SMP).

6.10 Clockwork RNN Models

In Chapter 1, the rationale for using models that operate at multiple temporal frequencies in stock time series was outlined. Such an approach is exemplified by the model proposed in [229]. Additionally, Ang and Lim's introduction of 'latent cross-attention learning between modalities of different time-scales and sparsity' [4] highlights the problem of sparsity, which is especially common in data with finer temporal resolution, such as minute-level granularity. Support for multi-frequency analysis is further provided by [43], where attention scores are analyzed across weekly and daily frequencies. Based on this analysis, the timing intervals for the computational model are experimentally determined and adjusted.

In this thesis, the CWRNN model, as described in [112], is employed. The implementation utilized herein is sourced from Github [2]; for implementation details, the reader is directed to the link in the footnote. In the original publication the

[2] https://github.com/ToruOwO/clockwork-rnn-pytorch

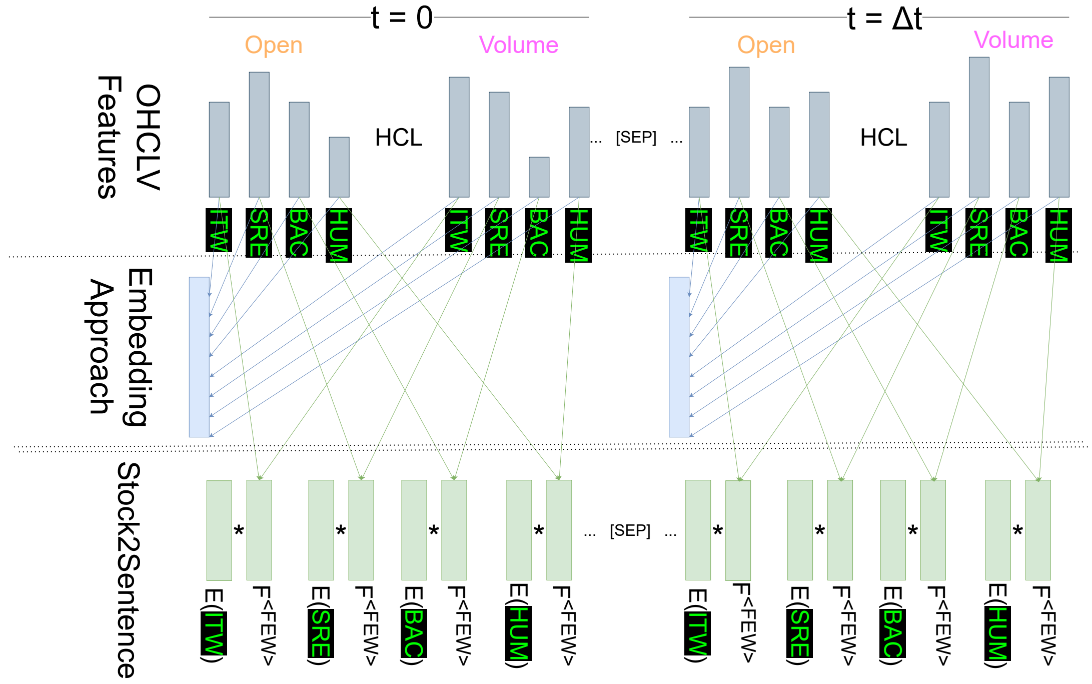


FIGURE 6.8: Sketch of the Stock2Sentence and embedding based approaches. The colors are also used in Listing 6.1.

CWRNN model, denoted as $\tilde{F}^{\langle \mathrm{M} \rangle}(.)$, ingests a tensor $\tilde{X} \in \mathbb{R}^{\tilde{\Delta} t \times \tilde{\xi}}$ representing audio data and subsequently generates an output $\hat{Y} \in \mathbb{R}^{\tilde{\Delta} t \times \tilde{\xi}}$.

Central to the architecture of the proposed CWRNN are its 'modules' $\mathbb{P}$, each of which is assigned a specific 'clock period'. Each module $p_i \in \mathbb{P}$ is set to its own temporal frequency, enabling different processing rates across modules. For the CWRNN experiments, the methodologies adopted from [141] are integrated as discussed in Section 6.4. These methods address the challenges of non-stationarity within the dataset, thereby stabilizing the training process and enhancing model reliability.

6.11 Proposed Adapted Speech Models

This section is mainly based on the authors publication [222].

Chapter 6 sets the goal of adapting the LLM pipeline to time-series stock price prediction, a central aim of this thesis. This builds on properties of contemporary LLMs that suit SF problems, as outlined in Chapter 1. The authors prior research, including proposed methodologies outlined in [222] and detailed in Section 6.6, demonstrates the feasibility of transforming multivariate time series data into a format to be processed by LLMs.

Initially, $F^{\langle \text{LLM} \rangle}(.)$ is delineated as encapsulating one among a quintet of prominent LLMs; specifically, BERT [40], LLaMA [216], GPT-2 [187], TransformerXL [31], and T5 [188]. Most models comprises an encoder component denoted as $\left(F_1^{\langle \text{E} \rangle}(.), \ldots, F_\rho^{\langle \text{E} \rangle}(.)\right)$, or a corresponding decoder component $\left(F_1^{\langle \text{D} \rangle}(.), \ldots, F_\rho^{\langle \text{D} \rangle}(.)\right)$. The subsequent use of the decoders follow the methodologies described in [224]. Initial explorations and discussions of these decoder-based approaches are deferred to Section 9.2.

As outlined in [222], refined methods are proposed to replace each of the three coarse-grained processing stages within $F^{\langle \text{LLM} \rangle}$: the speech model $F^{\langle \text{SM} \rangle}$, the embedding $F^{\langle \text{E} \rangle}$, and the tokenization $F^{\langle \text{TO} \rangle}$. In line with the definitions provided in Section 6.9, pretraining is framed as a model-specific adaptation designed for the ASMs. Regardless of the chosen approach, the model $F^{\langle \text{LLM} \rangle}(Z) = \mathbf{h}_{\text{CLS}}$ is defined using the strategy described for the $F^{\langle \text{BM-T} \rangle}$ in Section 6.5.

6.11.1 Embedding based Approach

In the initial approach, $\bar{X}$ is employed, whereby ξ is defined such that for $\xi \neq |C| \cdot \mathbb{F}$ can hold true. This definition enables the specification of ξ either as the original model size for $F^{\langle \text{LLM} \rangle}$ (for example, $\xi = 768$ for BERT) or as an optimized hyperparameter. The principal distinction between the standard transformer model as reported in [219] and the variants discussed in Section 6.8—excluding those with recurrent architectures—lies in the modifications tailored to LLMs. These modifications encompass specific implementations such as the choice of activation functions, the application of batch or layer normalization, and the utilization of either cross attention or merged attention mechanisms.

6.11.2 Stock2Sentence Approach

In the proposed Stock2Sentence method, the stock embedding from the S2V model is adapted in a way similar to how W2V embeddings may be used in LLMs in NLP.

When applied to an NLP context, the dimensionality of an embedded word vector $\dim(\tilde{e}^{(t)}) \propto |C|$ holds in all approaches discussed before. Furthermore, the specific temporal index of the word, $\tilde{v}^{(t)}$, at which the vector $\tilde{e}^{(t)}$ is positioned, aligns with t from $x_i^{(t)}$, representing the discrete temporal step in the stock sequence.

In the models presented in Section 6.8, an attempt was made to predict the market snapshot $X^{(t+1)}$. However, this analogy encounters limitations as the structural composition of NLP models does not use stacking of embedded word tokens, as observed in the proposed financial model.

Consequently, the proposed adopted methodology diverges from concatenating sequential market snapshots. Instead, each market state $X^{(t)}$ is treated as a sentence in the NLP sense.

If each market snapshot is conceptualized as an individual component of a comprehensive 'Text' X, the 'Sentence' construct from the domain of NLP can be adopted. To this end, a 'Sentence' $A^{(t)} \in \mathbb{R}^{\xi_{\text{S2V}} \times l}$ is defined, representing a market snapshot such that $A^{(t)} \equiv X^{(t)}$. Consequently, for the input $\mathcal{A} = \left[A^{(t)}, \mathbf{h}_{\text{PUNC}}, \ldots, \mathbf{h}_{\text{PUNC}}, A^{(t-\Delta t)}\right]$ holds.

The exact meaning of 'Sentence' and the dimension of l depend on the chosen methodological framework. In an effort to delineate distinct market snapshots, which are analogous to disparate sentences in textual analysis, inspiration can be drawn once again from NLP techniques. In NLP models, simple punctuation marks, such as the period ('.'), are commonly used to separate sentences.

In this context, a trainable token embedding $\mathbf{h}_{\text{PUNC}} \in \mathbb{R}^{1 \times \xi_{\text{S2V}}}$ is introduced, utilized specifically to separate individual market snapshots within the analytical model.

Untabulated experimental results suggest that the mere use of punctuation as delimiters is insufficient for generating a robust temporal structure in the analysis. To address this limitation, the concept of position encoding employed in transformer architectures is drawn upon. The integration of a learnable Embedding Matrix, $E_t \in \mathbb{R}^{\Delta t \times \xi_{\text{S2V}}}$, is proposed to enhance the temporal contextualization of data points.

Therefore $A^{(t)}[i, l] \leftarrow A^{(t)}[i, l] + E_t[t, i]$ is assigned. This adds a time-specific signal to each snapshot for temporal context. With the exception of TransformerXL and T5, the model's native position embeddings were used for all LLMs (meaning for example Rotary Position Embeddings were implicitly used for LLaMA). For TransformerXL/T5, they were implemented by the author. This means that all c_i in $A^{(t)}[j, i]$ have the same position embedding, as they are each semantically at the same position, i.e. the same time step t. Classic position embeddings (each input *vector* having a distinct embedding) were also tested, which led to performance losses in all cases.

Numerous non-Euclidean methodologies discussed in Section 3.0.6 incorporate dynamic graph structures that evolve temporally, identifying this adaptability as a critical factor for enhancing algorithmic prediction efficacy. In this approach, the dynamic nature of relationships between stocks is carefully addressed through three core mechanisms. First, the embeddings remain subject to continuous training rather than being statically frozen, as for example in [246]. Second, the temporal structure is inherently encoded through both positional and feature vectors, ensuring contextual coherence. Third, the embeddings undergo a proposed pretraining phase within the MPM, where they are further refined through fine-tuning, preserving contextual and temporal dependencies throughout the learning process. The experiments have shown that for all proposed tasks in the ASMs it is important to use the adapted normalization method of [168] from Appendix A.1.

Price Information Integration To systematically integrate spatial stock information with their corresponding temporal OHCLV features, the implementation of three distinct methodological approaches was explored.

The first approach involved using each feature to independently scale a distinct S2V vector. Empirical evaluations indicated that the addition of $\mathbb{F}$-specific S2V vectors and the significantly increased computational complexity subsequently degraded overall performance. Due to these limitations, this approach was deemed suboptimal and was not pursued further.

The second approach, presented in [222], involved scaling each OHCLV feature onto vectors of reduced dimensionality and subsequently stacking these vectors. However, this strategy also demonstrated inferior performance compared to the approach detailed in the following, and thus, was similarly discarded in favor of more effective methodologies.

The final approach proposed is based on having a learnable scaling vector based on the (OHCLV)-features. For this a linear layer $F^{\langle \mathrm{FEW} \rangle}(.)$ is defined with a learnable matrix $W_{\mathrm{FEW}} \in \mathbb{R}^{|C| \times \xi_{\mathrm{S2V}} \times \mathbb{F}}$ with an according bias $\mathbf{b}_{\mathrm{FEW}}$ and define the scaling vector $\mathbf{s}_i^t$ at t for a stock c_i as

$$\mathbf{s}_i^t = \mathbf{x}_i^{(t)} \cdot (W_{\mathrm{FEW}}[i])^T + \mathbf{b}_{\mathrm{FEW}}[i] \quad . \tag{6.63}$$

Further this 'Sentence' is defined as

$$A^{(t)} = \left[\mathbf{s}_i^t \star E[i], \dots, \mathbf{s}_{|C|}^t \star E[|C|]\right] \ . \tag{6.64}$$

One advantage of this proposed method is that default multi-head attention can now be used effectively since not all price information is in one position of the embedding dimension. In the investigations, an alternative representation of $\mathbf{x}_i^{(t)}$ or $\mathbf{e}$ in the form of a vector comprising various technical indicators was also considered. Specifically, beyond the standard OHLCV attributes, an extensive set of 204 additional indicators derived from the AV dataset, corresponding to 52 indicators per OHLC feature, was incorporated[3]. Despite this expansion, results were at best comparable—and often worse—than the OHLCV baseline, so it was not pursued further. This suboptimal performance is plausibly attributable to the substandard quality of the supplementary dataset, which exhibited a high prevalence of missing values. The Stock2Sentence transformation likely already captures the relevant market dynamics, leaving little gain from adding technical indicators mostly grounded in the OHCLV features.

S2V Order In a given instance $A^{(t)}$, the sequence in which the scaled S2V embeddings are arranged is hypothesized to be irrelevant, as these embeddings correspond to identical time steps. The order of the scaled S2V embeddings within one $A^{(t)}$ is systematically permuted in the experiments. Additionally, the introduction

[3] Indicators can be found in https://www.alphavantage.co/documentation/#technical-indicators

of a pretraining task described in the design is intended to improve the model's ability to identify semantically equivalent representations. Since the model shows no difficulty in recognizing that the pure order within t is irrelevant, and the pretraining task can be solved almost perfectly, no further investigations on this aspect are listed.

Prediction Heads Different prediction heads for SMP/SPP were also experimented with, following the idea of [79], where a distinct predictor is used for each stock. Instead of passing the (flattened) CLS embedding into the head, the learnable weigh $W_{\text{ASM-TP}} \in \mathbb{R}^{|C|\times 1\times \xi}$ is proposed with the corresponding bias to compute

$$\hat{\mathbf{y}}[i] = \left(\frac{1}{|J_i|}\right) \cdot \sum_{j\in J_i} \mathcal{A}^{(j)} \cdot W_{\text{ASM-TP}}[i]^T + \mathbf{b}_{\text{ASM-TP}} \tag{6.65}$$

with $J_i = \{i + (|C| + 1) \cdot l | \forall l \in \mathbb{N}_0 < \Delta t\}$. This gives each stock a dedicated prediction head with enough capacity to model its temporal and structural dynamics while using the processed spatial information of each stock.

6.11.3 Tokenization Approach

The final proposed methodology, as delineated in [222], involves the tokenization of numerical price data, i.e. the tokenization of regression values. This approach replaces the complete LLM pipeline by using the tokenization model $F^{\langle \text{TO} \rangle}$ and a defined vocabulary R. A fundamental distinction exists between NLP and SF; the former fundamentally addresses regression data, whereas the latter is a classification problem. All required tokens are included in R. Therefore

$$R = C \cup \{\text{‘-’}, \text{‘:’}, \text{‘[PUNC]’}\} \cup \{x | x \in \mathbb{N}_0, x \leq 10\} \cup \mathbb{F} \tag{6.66}$$

encompass all company identifiers as stock ticker, all numerical digits, interval feature identifiers denoted by $\mathbb{F}$, and other characters. Formally, $f : R \mapsto V \subset \mathbb{N}$ is performed through $F^{\langle \text{T} \rangle}$. With these tokens, the stock input can be tokenized, with $\mathcal{A}$ defined as the concatenation of $A^{(t)}$. Moreover, this tokenization allows any speech model to be applied directly, removing the need to skip the embedding phase, since the input is already provided in tokenized form.

The input $A^{(t)}$ is represented as a sentence by

$$A^{(t)} = \Big[c_1\ q_1\ \mathbf{x}_1^{(t)}[1], \quad c_1\ q_2\ \mathbf{x}_1^{(t)}[\mathbb{F}], \quad , \dots , \quad c_{|C|}\ q_{\mathbb{F}}\ \mathbf{x}_{|C|}^{(t)}[\mathbb{F}]\Big] \tag{6.67}$$

with $q \in \mathbb{F} \subset \mathbb{N}_0$ e.g. 'Open = 0'.

This input can now be tokenized. Each x_i^t is represented as

$$x_i^t[j] \leftarrow f_{\text{reverse}}(x_i^t[j] \cdot 10,000)\ . \tag{6.68}$$

The multiplication factor of $10,000$ is employed due to the database's convention of storing each price value with four decimal places. Volume values are max-min normalized for the other features the unscaled original values are used. This specific scaling ensures that each numerical value retains its precision post storage and retrieval. Furthermore, the reversal process delineated in Appendix A.8 serves to stabilize the training process. It standardizes the representation of price across different stocks by ensuring uniformity in digit placement: the first digit corresponds to the fourth decimal place, the second to the third, and so forth. This methodical reordering mitigates discrepancies in data handling and enhances the consistency of input features for subsequent analytical processes.

Larger numbers in the vocabulary are also experimented with to decrease the input length, i.e.

$$R^{(\theta_V)} = C \cup \{\text{`:'}, \text{`-'}, \text{`[PUNC]'}\} \cup \{x | x \in \mathbb{N}_0, x \leq \theta_V\} \cup \mathbb{F} \tag{6.69}$$

as well as using the respective speech model default vocabulary set $\tilde{R}^{(\text{<SM>})}$ (e.g. the BERT Vocabulary and Tokenizer). Both approaches were discarded: the first due to unsatisfactory results, and the second owing to excessively large inputs and poor performance, as also noted in [222].

```
[CLS]
 ITW 0: 0396  SRE 0: 0942  BAC 0: 0426  HUM 0: 4511 #...# Open
  # ...
 ITW 4: 0610  SRE 4:  0300 BAC 4:  0130 HUM 4:  0420 #...# Volume
[PUNC]  # Separator Token
  # ...
 ITW 0: 0696  SRE 0:  0742 BAC 0:  0226 HUM 0:  4511 # ...# Open
  # ...
 ITW 4: 0810  SRE 4:  0270 BAC 4:  0110 HUM 4:  0069 #...# Volume
[PUNC]  # Separator Token
[EOS]
```

Listing 6.1: Text representation of the tokenization based approach.

A summary of all approaches can be seen in Figure 6.8 (Stock2Sentence and Embedding based) and in Listing 6.1 (Tokenization based).

6.11.4 Expandability and Generalization

The latter two methodologies facilitate a notable enhancement in the expandability of the model. This adaptability is crucial given the dynamic nature of stock markets, where companies may either enter by listing/IPOs or exit due to delisting or bankruptcy.

Particularly with datasets of higher temporal resolution, such as those measured at minute intervals, it is frequently observed that data may not be consistently available for each t for all companies i.e. $\exists i \in \mathbb{N} : \exists t \in \mathbb{N} : \nexists x_i^t$. To address these instances of incomplete data, various padding methodologies have been previously delineated.

In the context of SF models, employing inputs such as X or $\bar{X}$, particularly in scenarios devoid of a S2V embedding, presents certain limitations. These models lack extensibility primarily because ξ, which is directly derived from C, depends on the position of each c_i within the model structure. Unlike the Stock2Sentence and Tokenization, the conventional methods do not facilitate the straightforward integration or exclusion of companies based on the availability of data during each training step.

Furthermore, as outlined in Chapter 1, the models' ability to expand and generalize is of importance. The adoption of the proposed S2V representations enhances the model's ability to incorporate new stocks. This characteristic not only addresses the mentioned limitations but also supports faster learning of interrelations among data points, thereby improving the model's overall effectiveness and adaptability. The correlation and relationship learning is further motivated by many studies, including [258], have argued against the use of fixed correlation information.

6.11.5 Pretraining

As noted in Section 6.9, most language models are pretrained with MLM and NSP. These methodologies are applicable to the training of ASMs as well, albeit with requisite modifications to better suit the specifics of the Stock2Sentence framework.

Next-Sequence-Prediction as Trend-Matching The methodology outlined in Section 6.9 can, in principle, be directly extended to the NSP task in the following manner: Within $\mathcal{A}$, a designated marker token [TM] embedding is inserted at predefined positions. Subsequently, segments of stock price histories are concatenated before and after this marker token.

To enable the model to effectively differentiate between these segments, a learnable embedding matrix $E_{\text{TM}} \in \mathbb{R}^{2 \times \xi_{S2V}}$ is used, which is added to each $\mathbf{e}$ based on whether it belongs to the sequence preceding or succeeding the [TM] token. This proposed approach parallels the segment embedding mechanism employed in many LLMs, such as BERT [40]. Finally, the model is tasked with performing a binary classification to determine whether the two adjacent stock price history segments exhibit sequential continuity.

As illustrated in Section 7.5.2, this task presents significant challenges, necessitating the exploration of various implementation strategies, normalization modules, loss functions, and activation functions. These strategies include processing the folded sequences, or incorporating the processed positional information of the CLS token within the prediction head.

An alternative task is designed to move beyond a binary decision task, wherein, during each training step, the entire trend is not simply replaced with another trend from the mini-batch. Instead, a specific $c_i \in C$ is selected per batch and swapped with the trend positioned after the CLS token, requiring the model to classify the corresponding stocks identity. Partially flattening each sequence into a separate head with a 2 dimensional output and a cross-entropy/softmax decision head was also tested, but this design was abandoned.

Masked-Language Modeling as Masked Price Matching In Section 6.9, the MLM task has been previously modified by selectively masking the temporal or the spatial/market axis. However using Stock2Sentence and $\mathcal{A}$, the horizontal dimension is comprised solely of embeddings, rather than actual price data as in Section 6.9.1, where the representation of prices varies. This configuration allows the use of methodologies that are more closely aligned with those applied in NLP. Again a mask is defined (this time only cloze type masking as done in all the LLMs) as $M \in \{0,1\}^{\dim(\mathcal{A})}$. As in timestep masking in Section 6.9.1 $\forall i \in \mathbb{N} < \Delta t : \mathbf{b}[i] \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\nu_M)$ with $\dim(\mathbf{b}) = \dim(\mathcal{A})[1]$ and $M[i,j] = \mathbf{b}[j]$ holds.

This procedure can be adapted for SMC without requiring significant modifications to the LLM architecture. First, it remains debatable whether using a uniform [MASK] token embedding across all stocks is appropriate, given that the model lacks any specific indicator of the stock to predict.

If the same C is used for each training step, the same c_i consistently appears in the same location within $A^{(t)}$, enabling the model to potentially learn that all $c_i \in C$ occur between two [PUNC] token embeddings once but this structural knowledge does not benefit SF problems. As mentioned in Section 6.11.2 and shown in the experiments in Section 7.7, the model has no problems mapping the position of the vector in the input sentence to the stock.

As the position of a scaled embedding within $A^{(t)}$ lacks inherent semantic meaning a different procedure is required if C changes in each time step. Following Section 6.9.1, a token c_i is masked by using its unscaled embedding $E[i]$ (or $-1 \cdot E[i]$ to indicate a masked variant). Subsequently, these obfuscated tokens are assimilated into a linear layer analogous in functionality to the MLM.

This layer is denoted as Masked Company Price Modeling ($F^{\langle \text{MCPM} \rangle}$), a linear map with weights $W_{\text{MCPM}} \in \mathbb{R}^{\xi_{\text{Stock2Vec}} \times (|C| \cdot 2)}$. The main goal of this layer is to perform SMC on the masked price $\mathbf{x}_i^{(t)}$. This approach bears similarity to the S2V-vocabulary-based methodology discussed in Section 6.6.

Experiments were conducted to explore alternative architectures, employing a dual-layer design: one layer dedicated to predicting the SMP label and another for c_i (i.e. utilizing $W_{\text{MCM}} \in \mathbb{R}^{\xi_{S2V} \times |C|}$). However, this design had challenges when attempting to jointly train the two objectives. Furthermore, some uncertainty remains about whether the model fully understands the underlying task requirements, calling into question its ability to handle both prediction goals at once. The disadvantage of the vocabulary based approach is that no SME can be carried out in this way.

Text Corpus Adaption Due to the quadratically escalating computational time and memory complexity inherent in the ASMs, it is not possible to incorporate the complete set of stocks C, into the pretraining phase. This limitation holds true not only for C = S&P-500 🇺🇸, but also for more extensive datasets like All$^{(2010:)}$. An alternative methodology was explored to address this, wherein a varying subset $\dot{C}^{(i)} \subset C$ is sequentially integrated at each training step i for both MPM and TM. A conceptually similar approach is introduced in [189] with the multi-transformer model, which, at each training step, selects a distinct random subset of the training data to enhance learning dynamics.

Given the architectural extensibility of ASMs and the transmission of spatial information via S2V representations, it was conjectured that ASMs could be effectively trained using different $c_j \in \dot{C}^{(i)}$ at successive intervals. Such an approach allows for the incorporation of more granular representations of both S2V and broader market dynamics into the model. As a result, the learned representations extend beyond $c_j \in \dot{C}$ to $c_j \in C$.

Implementing this approach involves several key requirements. To maintain generalizability across different datasets, the output dimension of the MPM head must necessarily align with the cardinality of the complete stock set $|C|$. Furthermore, experiments showed that the computation of loss for positions not under consideration requires a masking strategy. The loss calculation must be modified as

$$f_{\text{softmax}}(\breve{\mathcal{H}}(\hat{\mathbf{y}}[i], \mathbf{y}[i])), \forall c_i \in \dot{C}^{(t)} \tag{6.70}$$

to prevent the overestimation of loss from unutilized positions, which could otherwise lead to training instabilities.

In addition to these modifications, experiments were conducted with hierarchical softmax, negative sampling, and adaptive softmax techniques. Unfortunately, these methods did not yield successful outcomes in enhancing the training efficacy or model performance.

It was observed that training slows markedly when all elements of C can be selected at each timestep t. This observation will be elucidated further in Sections Chapter 7 and Chapter 8. The findings suggest that the critical factor is not merely the number of potential elements c_i, but rather the combinatorial possibilities $\binom{|C|}{|\dot{C}|}$ (or $\binom{|C|}{|\dot{C}|} \cdot \mathbb{T}$) for the creation of the subset $\dot{C}$ i.e. represent the number of relationships between c_i that the model has to learn.

Two methodological approaches have been explored to address this limitation. Firstly, C is redefined such that $|C| - \epsilon = |\dot{C}|$. Secondly, to incorporate more c_i within the training process, $\acute{C} = \{\dot{C}^{(j)}\}_{j=1}^{\zeta}$ is defined, with ζ being a pre-specified hyperparameter. Consequently, for each training step j, $\dot{C}^{(j)} \sim \mathcal{U}(\acute{C})$ applies.

Other experiments have been carried out, including partial masking and company masking as in Section 6.9.1. However, these were not pursued further due to the poor performance in the experiments. Since Δt and $|C|$ must be relatively restricted for training due to the size of the ASMs, the problem here probably lies in the insufficient context data.

Another masking approach that had to be discarded was the classic MLM of tokenization-based approaches with the original LLM tokenizer on the raw `.json` data of the AV dataset. This achieved an MLM performance around 50% and hardly helped in finetuing.

As already mentioned in [217], the embeddings trained in MPM are context-sensitive due to their adaptation of LLM MLM and the omission of frozen model parts. To the authors knowledge, these proposed embeddings are the first context-sensitive stock embeddings.

Chapter 7

Results

For the discussion of the results, the structure outlined in Figure 1.2 and how it will be discussed in Chapter 8 is followed. This chapter presents the empirical findings following the structure in Figure 1.2, aligned with the methodology of the preceding chapters. It starts with establishing baselines, then moves to embedding-based and (recurrent) transformer-based methods, and finally explores pretraining and contextualized representations.

The first Section 7.1 reports the baseline results. Naive models and optimized grid-search configurations are evaluated across different temporal resolutions and datasets. These results provide essential benchmarks against which the subsequent, more complex models can be compared.

Section 7.2 introduces the results of the Stock2Vec models, which adapt the Word2Vec paradigm to financial time series. The analysis is divided into two parts: first, the extrinsic performance of the embeddings in SMC, SPE is assessed; second, the intrinsic quality of the embeddings is examined through similarity analysis, concept categorization, outlier detection, and visualization. This section shows how embedding-based representations capture structural information in the data.

Building on this, Section 7.3 investigates the CWRNN architecture, representing a recurrent approach to temporal modeling. The results highlight the strengths and limitations of hierarchical mechanisms in comparison to both baseline and later discussed models, particularly with respect to temporal granularity.

Section 7.4 presents the results of the QMSEs, an adaptation of the Doc2Vec framework. Here, the focus is on both reconstructive and representative performance, with special attention given to the ability of QMSEs to encode complex market dynamics and identify anomalies. Intrinsic evaluation further illustrates their representational capacity and limitations.

Section 7.5 examines the transformer-based architectures. The experiments focus on different pretraining strategies, sequence lengths, masking approaches, and context configurations. The analysis investigates whether pretraining improves downstream performance and stability and how transformers handle long temporal sequences.

The chapter concludes with the presentation of results from the ASMs in Section 7.6 (Embedding Based Approach), Section 7.7 (Stock2Sentence), (Tokenization Based Approach) Section 7.8. These models integrate contextualized embeddings and transformer mechanisms into a unified framework and are evaluated with respect to both pretraining and downstream tasks. The section evaluates how ASMs support pretraining, finetuning, and expandability across datasets, and relates the results to the research questions.

7.1 Baseline Results

After the required hyperparameters were found via try-and-error, a grid search was conducted to further optimize them. The parameters examined were, $\Delta t = \{7, 14, 21\}$; $E_K = \{\text{True}, \text{False}\}$; $E_C/E_{\mathbb{F}} = \{\text{True}, \text{False}\}$. For interday evaluations, ETFs were also considered as supplementary data. The results for each dataset are tabulated in Table 7.2 for the SPP model and in Table 7.3 for the SMP model. The performance metrics of the naive SPP model (Section 6.2), evaluated across the training, validation, and test datasets, are documented in Table 7.1.

For all experiments, three temporal resolutions were considered, each derived from OHLCV features aggregated at the respective interval. Temporal resolution is defined as the granularity of the time series, from long intervals that smooth fluctuations (interday) to short intervals that capture rapid dynamics (1min). Stock

Table 7.1: Overview of the expected values for SPP according to naive models for each dataset.

Model	Interval	sMAPE ↓	MAPE ↓
Naive	Interday	2.033/2.245/2.004	2.034/2.248/2.003
Naive	60min	0.841/0.847/0.669	0.876/0.763/0.628
Naive	1min	0.093/0.092/0.095	0.089/0.088/0.091

market data at multiple granularities are provided by the AV API, enabling systematic evaluation under different temporal conditions. Among the available options, three representative resolutions were selected: the interday setting (e.g. end-of-day values for the closing price), which emphasizes long-term market movements while smoothing short-term fluctuations; the 60-minute interval, which reflects intraday trading patterns with a balanced trade-off between noise and broader signals; and the 1-minute interval, which captures high-frequency dynamics with maximal granularity, albeit at the cost of increased stochasticity and noise.

Table 7.2: SPP performance of the best baseline models on each dataset.

Model	Interval	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
$F^{\langle \text{BM-R} \rangle}_{+E_K}$	Interday	1.655/1.401	1.657/1.403	0.504/0.501	0.005/0.000	0.490/0.496
$F^{\langle \text{BM-R} \rangle}$	60min	0.422/0.369	0.423/0.369	0.503/0.503	0.005/0.005	0.490/0.489
$F^{\langle \text{BM-R} \rangle}$	1min	0.062/0.062	0.062/0.062	0.501/0.501	0.002/0.002	0.462/0.465
$F^{\langle \text{BM-T} \rangle}$	Interday	1.658/1.398	1.659/1.400	0.497/0.501	-0.005/0.003	0.510/0.503
$F^{\langle \text{BM-T} \rangle}$	60min	0.446/0.388	0.446/0.388	**0.507/0.505**	0.015/0.010	0.498/0.498
$F^{\langle \text{BM-T} \rangle}$	1min	0.073/0.070	0.073/0.070	0.502/0.502	0.003/0.003	0.482/0.483
$F^{\langle \text{BM-L} \rangle}_{+C/\mathbb{F}+E_K}$	Interday	1.660/1.402	1.662/1.404	**0.504/0.501**	0.002/0.001	0.541/0.531
$F^{\langle \text{BM-L} \rangle}$	60min	0.412/0.361	0.413/0.361	0.503/0.503	0.005/0.005	0.488/0.492
$F^{\langle \text{BM-L} \rangle}$	1min	0.0816/0.365	0.0816/0.365	**0.503/0.502**	0.006/0.003	0.491/0.493

Table 7.3: SMP performance of the best baseline models on each dataset.

Model	Interval	Acc ↑	MCC ↑	F1 ↑
$F^{\langle \text{BM-R} \rangle}_{+\text{C}/\mathbb{F}}$	60min	0.520/0.521	0.007/0.012	0.362/0.358
$F^{\langle \text{BM-R} \rangle}_{+\text{C}/\mathbb{F}}$	1min	0.508/0.513	0.014/0.017	0.465/0.459
$F^{\langle \text{BM-T} \rangle}$	Interday	**0.500/0.502**	-0.000/0.003	0.517/0.520
$F^{\langle \text{BM-T} \rangle} + F^{\langle \text{R} \rangle}$	60min	**0.527/0.527**	0.006/0.006	0.250/0.256
$F^{\langle \text{BM-T} \rangle}$	1min	**0.540/0.543**	0.046/0.050	0.428/0.423
$F^{\langle \text{BM-L} \rangle}$	Interday	0.504/0.501	0.008/0.002	0.493/0.496
$F^{\langle \text{BM-L} \rangle}$	60min	0.514/0.511	0.020/0.025	0.469/0.499
$F^{\langle \text{BM-L} \rangle}$	1min	0.516/0.532	0.040/0.051	0.488/0.467

7.2 Stock2Vec Results

The S2V results are listed in the following. All experiments were done on the interday datasets due to hardware limitations and the long training time of the S2V models. The runs of models $F^{\langle \text{C-*-\{SG, CBOS \}} \rangle}$ as defined in Section 6.6 are repeated five times.

Evaluation Embeddings are evaluated along two dimensions. The first dimension of this analysis, termed ‘model evaluation’, compares model performance on the SMC/SPE tasks. Detailed discussion at this results is rather uncommon in NLP. The second dimension is concerned with the embeddings’ intrinsic properties, including semantic content and usefulness as representations. Several evaluation methods are available in NLP for this second dimension.

Nevertheless, the scholarly community continues to grapple with establishing a unified set of quality criteria for the assessment of word embeddings, as highlighted in [227]. Adhering to the methodologies delineated in [196], the evaluation of embeddings can be segmented into two distinct approaches. The first approach involves ‘Extrinsic Evaluators’ [196], where the embeddings are applied to downstream tasks, as exemplified in Section 7.7. This application facilitates performance comparisons via empirical metrics. [167] criticizes the common practice of evaluating relationship graphs (in this case embedding based relationship representation) based on their performance in downstream tasks, as this can be misleading since robust models may mask deficiencies in the graph, and the resulting graphs often exhibit limited generalizability.

The second approach utilizes ‘Intrinsic Evaluators’ [196], which concentrate on analyzing the inherent characteristics of the embeddings themselves. According to Wang et al. [227], intrinsic evaluations fall into two categories. The first category encompasses ‘absolute’ scores, which are quantitatively computed and subsequently compared across various embeddings. The second category comprises comparative evaluations, which rely on subjective human assessments of the embeddings’ quality. These methodologies are illustrated in Figure 7.1.

In the following, the framework of Wang et al. is followed. The focus in this section will predominantly be on model evaluation and the intrinsic analysis of embeddings. Extrinsic evaluations are presented with the respective models in Section 6.11.

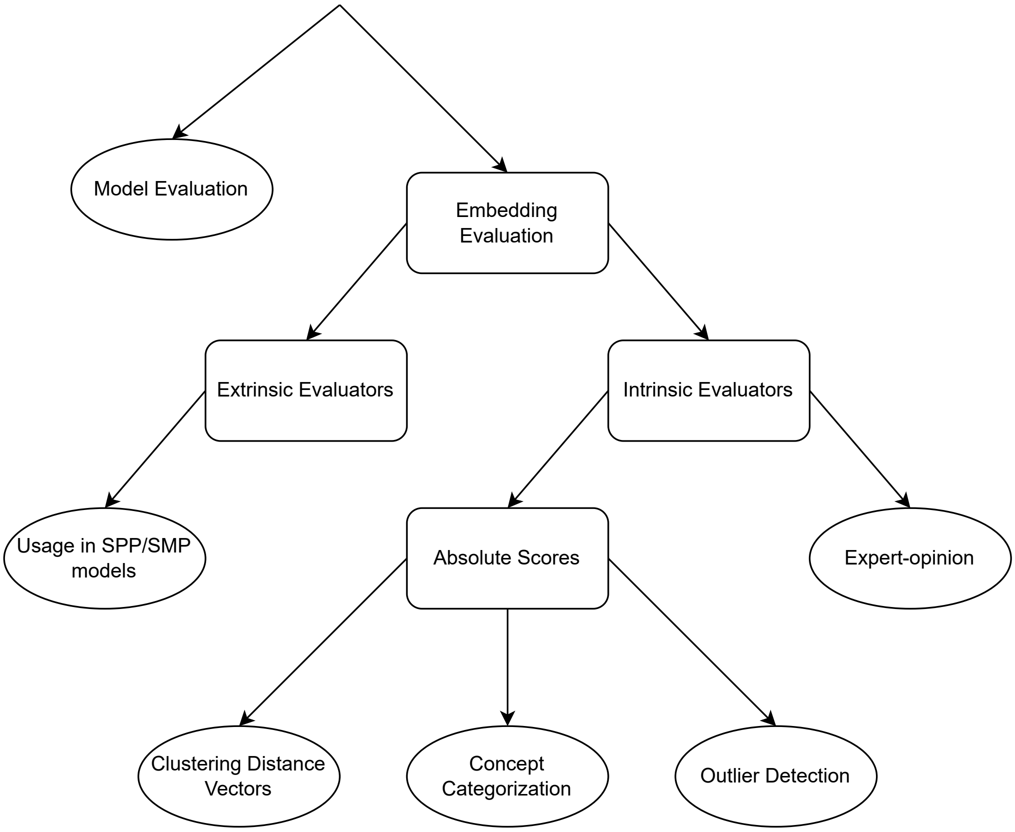


FIGURE 7.1: Schematic representation of embedding/S2V evaluation methods.

7.2.1 Model Evaluation

The performance of the S2V models in classification and regression tasks does not necessarily serve as a definitive indicator of embedding quality. However, the evaluation of these tasks remains the primary focus. Enhanced performance in these tasks suggests the successful identification of a method capable of representing stocks as high-dimensional vectors. The efficacy of this method is predicated upon the discriminative capacity of the spatial features within the stock vector representation. These features must accurately identify individual stocks, encapsulate the current market dynamics through the integration of temporal price information, and delineate the interrelations among stocks by leveraging their spatial characteristics. As delineated in Section 6.6, the option to incorporate either the market axis or the temporal axis into the S2V task is presented. The former is of particular interest as it facilitates the creation of embeddings that can be analyzed in relation to other stocks. Next to the S2V models, the context-sensitive embedding derived from a (T5-based) ASM model in Section 7.7 ([6]) is used.

Market Axis Following Section 6.6, SMC and SPE are used to generate embeddings. The CBOS-C models conceptually address the classification of stock movements and prices. This differs fundamentally from the predictive models SMP/SPP in that

$$F_{\Theta}^{\langle \text{C-CBOS} \rangle} : \mathbb{M}(t) \rightarrow x_i^{(t)} \tag{7.1}$$

is calculated where the current market $\mathbb{M}(t) = \mathbf{x}_j^{(t)} : \forall c_j \in C \setminus \{c_i\}$ is mapped to the features of a specific stock at the same time step.

Despite the domain's intrinsic difficulty (e.g. in SMP/SPP), S2V models demonstrate strong performance on the closely related SMC task. The outcomes for the CBOS-C models are presented in Table 7.4. The models exhibit variable performance, with accuracy metrics for movement classification attaining levels up to 80%, contingent upon both the specific features employed and the prevailing market. Further discussion of SPE is omitted because little learning was observed. The SPE models were not effective enough to warrant detailed analysis.

An adaptation was made to the CBOS-C predictive setup to compare SMC with SMP. Specifically, the target was modified to $y = x_i^{(t+1)}$, diverging from the conventional $y = x_i^{(t)}$. A schematic representation of this predictive S2V model is depicted in Figure 7.2. As expected, performance of the simple model was poor. Notably, the performance was not above a naive or random baseline. Given that the prediction task is constrained to forecasting $t + 1$ returns based solely on returns at t, the observed outcomes are not entirely unexpected. The SMC results are particularly noteworthy when considering the inherent simplicity of the CBOS-C models and their minimal parameterization. These models are comprised of a single computational layer and one activation function, supplemented only by scaled embeddings.

Performance may improve with more complex architectures. Specifically, the trend indicating that larger values of ξ correspond to slightly improved accuracy suggests avenues for further optimization. This line of investigation is extended in the ASMs in Section 7.7.

Information on price levels and temporal fluctuations is incorporated by using returns. As shown in Table 7.4, using SMP labels to represent price information performs comparably to, and sometimes slightly better than, logarithmic returns. Vocabulary-based methods are comparable to, and in some cases exceed, alternative approaches. Two main points can be drawn. Firstly, the S2V models exhibit a robust capability to discern and classify movements of individual stocks, as opposed to merely analyzing aggregate market dynamics. This suggests that the spatial representations are sufficiently discriminative to encode inter-stock relations. Secondly, the effectiveness of the NLP-W2V approach is shown, highlighting its suitability for analyzing stock market dynamics.

In the context of the SPE task within the C-CBOS framework, no hyperparameter configuration was identified that enabled effective learning. Even attempts to deliberately overfit the training data proved unsuccessful, despite an extensive search over different learning rate values and $\xi \in \{48, 128, 768, 1024\}$. The model exhibited a brief reduction in MSE loss during the initial epochs, after which it became trapped in a suboptimal solution. This stagnation was consistent across all configurations tested. Furthermore, the RMSE and MSE metrics showed minimal variation across models, with the RMSE experiencing a maximum reduction of approximately 35% relative to its initial value.

The observed limitations are likely attributable to the model's simplicity, which appears insufficient to accurately map returns to precise regression values. Nevertheless, these findings should not be interpreted as undermining the overall suitability of the S2V model. On the contrary, the model demonstrated highly satisfactory performance in the SMC task, underscoring its potential in alternative applications.

For C-SG approaches employing SPE as the target, the results exhibit a similar trend. Further details are omitted because the results follow the same pattern.

The SG approach presents significantly greater challenges, as evidenced by its comparatively weaker performance. Conceptually, this approach attempts to classify the collective behavior of the entire market based on the returns of a single stock, i.e.

$$F_{\Theta}^{\langle \text{C-SG} \rangle} : \mathbf{x}_i^{(t)} \rightarrow \mathbb{M}(t) \quad . \tag{7.2}$$

Performance is limited, likely because insufficient information is available from a single stock to predict the entire market. Efforts to improve this framework by targeting a subset of the market, $\mathbb{M}' \subset \mathbb{M}$, were explored through alternative model configurations. However, these attempts did not yield satisfactory results, further highlighting the limitations of the SG approach in this context.

TABLE 7.4: Performance of the C-CBOS SMC models.

Model	ξ	Target	C	Acc ↑
CBOS + RLR	128 [1] / [2]	Close	S&P500 🇺🇸	0.701/0.710
CBOS + SMC labels	128	Close	S&P500 🇺🇸	**0.702/0.721**
CBOS + RLR	128	Volume	S&P500 🇺🇸	0.650/0.641
CBOS + RLR	128	OHCLV	S&P500 🇺🇸	0.686/0.696
CBOS + SMC labels	128	OHCLV	S&P500 🇺🇸	0.690/0.702
CBOS + RLR	768	Close	S&P500 🇺🇸	0.701/0.717
CBOS + RLR	768	OHCLV	S&P500 🇺🇸	0.680/0.697
CBOS + Vocab based [3]	128	OHCLV	S&P500 🇺🇸	0.687/0.698
CBOS + RLR	128	Close	All$^{(2010:)}$	0.655/0.658
CBOS + SMC labels	128	Close	All$^{(2010:)}$	0.656/0.670
CBOS + RLR	128	OHCLV	All$^{(2010:)}$	0.716/0.703
CBOS + SMC labels	128	OHCLV	All$^{(2010:)}$	**0.724/0.723**
CBOS + RLR	128	Close	All	0.644/0.682
CBOS + Vocab based [4]	128	OHCLV	All	**0.755/0.698**
Predictive + RLR	128	Close	S&P500 🇺🇸	0.521/0.490

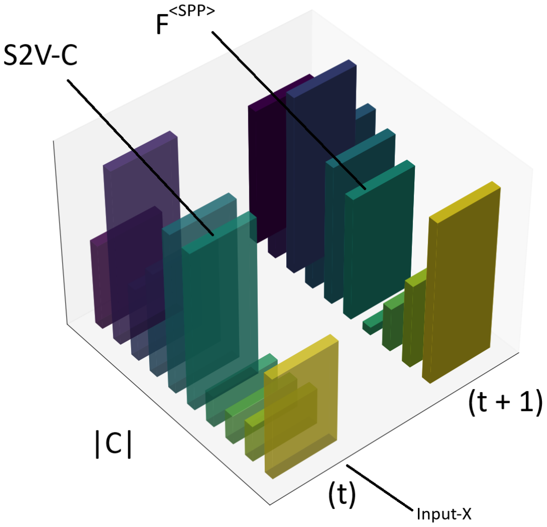


FIGURE 7.2: Schematic illustration of the comparison of SMP/SPP (predictive) and SMC/SME (classification/estimation) models. The X axis represents the different stocks on the market axis, the Y axis the time slice on the temporary axis and the Z axis the stock prices.

Table 7.5: Performance of the X-CBOS SMC models.

Model	ξ	Target	ϖ	C	Acc ↑
CBOS + RLR	128	Closing Price	20	S&P500 🇺🇸	0.526/0.506
CBOS + SMC Labels	128	Closing Price	20	S&P500 🇺🇸	0.516/0.503
CBOS + SMC Labels	128	All Features	20	S&P500 🇺🇸	0.564/0.565
CBOS + RLR	128	Closing Price	40	S&P500 🇺🇸	0.527/0.506
CBOS + RLR	128	Closing Price	100	S&P500 🇺🇸	0.522/0.503
CBOS + RLR	128	Volume	20	S&P500 🇺🇸	0.605/0.598
CBOS + RLR	128	All Features	20	S&P500 🇺🇸	**0.697/0.714**
CBOS + RLR	768	Closing Price	20	S&P500 🇺🇸	0.526/0.508
CBOS + RLR	128	Closing Price	20	All$^{(2010:)}$	0.637/0.763

Temporal Axis On the temporal axis, CBOS performance is substantially lower than for market-axis variants (see Table 7.5). This observation substantiates the significance of considering intercorrelations among stocks, a notion consistently emphasized by various researchers in Chapter 2. Consequently, this casts doubts on the feasibility of univariate time series forecasting within the stock domain. An approximation of CBOS-C performance is obtained only when all OHCLV features are used. Although performance drops noticeably when using SMC labels instead of RLR data, it still surpasses that of other methods. This underscores the importance of indicator correlation, previously highlighted in [62], and sets the expectations for the performance in the masking task. An exception is noted in the prediction of trading Volume, which was identified in [223] as markedly more predictable. Additionally, the model convergence is notably quicker in this instance. Successful learning is generally not achieved with smaller ϖ values, which is relevant for the later masking tasks. As with other CBOS-C trials, no feasible method was identified to successfully execute an SME task or to train an SG-X model. These results are likely due to the limited expressive power of simple models and the low informativeness of individual returns.

C/X-Models In Table 7.6 it can be seen that training on both axes improves the overall performance.

Table 7.6: Performance of the X/C-CBOS SMC model.

Model	ξ	Target	ϖ	C	Acc ↑
CBOS + RLR + X/C [5]	128	All Features	20	S&P500 🇺🇸	0.800/0.802

The following conclusions can be drawn from the analysis presented:

- Scaled embeddings are effectively suited for representing market scenarios, affirming their applicability in capturing the essence of market dynamics.

- The classification of stock price movements, based primarily on market data, is feasible with a relative simplicity, provided that sufficient information about price fluctuations is available. This indicates that basic market-driven factors play a significant role in stock price classification.

- A considerable level of confidence in the quality of these embeddings is warranted. Although these embeddings may not provide strong insights into industry clusters, country clusters, or intuitive stock similarities—as previously reported in other studies—they help in understanding relationships between stocks.

The ease of learning these relationships suggests potential gains in predictive tasks. By leveraging relationship data, the model may extend the predictability observed in certain stocks to others, thereby improving overall predictive accuracy (as discussed in Figure 8.1).

7.2.2 Embedding Evaluation

In subsequent chapters, the application of models in downstream tasks, such as ASMs, as extrinsic evaluators, will be examined. The following sections focus on intrinsic evaluation. To this end, five plus one distinct embeddings have been selected for detailed analysis (with the T5 ASM embeddings):

- [1] CBOS-C-SMC run for Closing Price (S&P-500 🇺🇸)

- [2] CBOS-C-SMC run for Closing Price scaled at random timestep and with random missing stock (ROST 🛒 at (2016-06-28)) (S&P-500 🇺🇸)

- [3] CBOS-C-SMC run with all OHCLV features (S&P-500 🇺🇸)

- [4] CBOS-C-SMC run with all OHCLV features on (All$^{(2010:)}$ dataset)

- [5] CBOS-X/C-SMC (S&P-500 🇺🇸)

- [6] Context sensitive ASM embeddings

7.2.3 Absolute evaluation methods

Absolute evaluation methods from W2V and their adaptation to SF are summarized below. Modifications to the metrics/approaches needed for stock data are outlined. Mikolov et al. have developed the Semantic Syntactic Word Relationship test, as documented in [156]. This method relies on a basic property of the embedding technique, which assumes that if $\tilde{v}^{(i)}$ and $\tilde{v}^{(j)}$, as well as $\tilde{v}^{(u)}$ and $\tilde{v}^{(w)}$, are related in a certain way, then a similar relationship $\tilde{\mathbf{e}}^{(i)} - \tilde{\mathbf{e}}^{(j)} \approx \tilde{\mathbf{e}}^{(u)} - \tilde{\mathbf{e}}^{(w)}$ should appear. Now $(\tilde{\mathbf{e}}^{(i)} - \tilde{\mathbf{e}}^{(j)}) + \tilde{\mathbf{e}}^{(u)}$ is calculated, hoping that $\nexists \tilde{\mathbf{e}}^{(t)} : \tilde{\mathbf{e}}^{(t)} \neq \tilde{\mathbf{e}}^{(w)} : (\tilde{\mathbf{e}}^{(i)} - \tilde{\mathbf{e}}^{(j)}) + \tilde{\mathbf{e}}^{(u)} - \tilde{\mathbf{e}}^{(t)} < (\tilde{\mathbf{e}}^{(i)} - \tilde{\mathbf{e}}^{(j)}) + \tilde{\mathbf{e}}^{(u)} - \tilde{\mathbf{e}}^{(w)}$ holds true. The utility of a similar evaluative approach is also reported in [227] and [195]. The efficacy of these methodologies is assessed through the preparation of sets comprising word pairs that are either semantically or syntactically associated.

Adapting NLP evaluation methods to SF is challenging because there is no standard, objective evaluation set criteria exists. Notably, prevalent techniques such as the Word Analogy Task delineated in [227] and the Word Similarity Task, in [227] and [63], prove problematic for adaptation. This difficulty primarily arises due to the indeterminate relational dynamics between stocks and the lack of distinct properties by which they can be systematically categorized and assessed. However in [196], a stock similarity task is demonstrated using JPM 🪙 as a reference entity. This example produces a result in which other finance-related stocks are identified as similar entities, illustrating a possible approach for sector-specific evaluation. Analogical inference is also used in [196], combining embedding properties with expert opinions. The similarity analysis was repeated as a test using JPM 🪙 and [1] for evaluation. The results are presented in Table 7.7, which details the most similar companies based on maximum-minimum normalized distances. Additionally, Table 7.8 provides an assessment of the companies exhibiting the highest cosine similarity values. The findings indicate that the entities identified through distance-based metrics predominantly belong to the financial sector; however, the distinction is less pronounced compared to the results obtained in [196]. Notably, many of the identified financial companies are concentrated within the insurance sector, diverging from the outcomes in [196], where the most similar entities included ‘classic’ financial institutions such as GS 🪙, C 🪙, and MS 🪙.

In Table 7.9, a different test on SCHW 🪙 is presented with [6]. Overall, the distances are significantly different from one another compared with the S2V embeddings. On the other hand, stocks from the financial sector mainly appear in the top 10. It is interesting to see that there are also more indicative relationships, such as the proximity to PAYX 🛒. SCHW 🪙 is currently (as of 25.3.2025) with 2.79% one of the largest institutional shareholders of PAYX 🛒 which establishes a relationship between the two[1].

Table 7.7: Stock similarity example using smallest embedding distances.

c_i	Distance
HBAN 🪙	0.744
AON 🪙	0.746
REGN 🧬	0.754
BRK.B 🪙	0.765
IT 🛒	0.766
UDR 🏠	0.767
TER 🧬	0.768
F 🏭	0.774
EQT ⚡	0.778

Table 7.8: Stock similarity example using cosine similarity.

c_i	Cosine similarity
TAP 🏭	0.287
WHR 🏭	0.211
RSG ⚡	0.210
BA 🏭	0.203
ADSK 💡	0.199
PCAR 🏭	0.196
GILD 🧬	0.195
MCD 🛒	0.188
ADP 💡	0.188

Table 7.9: Stock similarity using cosine similarity for context sensitive embeddings trained using [6].

c_i	Cosine similarity
PAYX 🛒	0.218
PNC 🪙	0.212
HBAN 🪙	0.192
ROL 🛒	0.190
AFL 🪙	0.161
FITB 🪙	0.150
EFX 🛒	0.148
UNH 🪙	0.127
ADSK 💡	0.124

[1] http://bit.ly/41ZR9U5

More intuitive results are sometimes obtained with scaled spatio-temporal embeddings ([2]). For instance, on a randomly selected day, the most similar stocks to MSFT 💡, as identified by the approach, were EBAY 🛒, AMD 🏭, and AFL 🪙. This outcome fits better with one would expect.

It is noteworthy that the majority of existing research on stock embeddings predominantly employs methodologies that incorporate industry and sector classifications as integral components of the evaluation process. Furthermore, many of these approaches are explicitly designed to map stocks based on such classifications from the outset (i.e. work top-down), presumably because these criteria represent the most accessible and objective benchmarks within the domain of financial markets. This work is based on performance observed in model and auxiliary-task evaluations. Intrinsic evaluation is included for completeness and comparability. Unlike traditional approaches that prioritize the structured organization of stocks, the focus is directed towards the development of spatial embeddings with the possibility to integrate temporal (i.e. price) features.

A significant challenge inherent to the authors approach, particularly in its application ASMs, is the high dimensionality of the embeddings. This issue is accentuated when compared to methodologies such as those described in [196] [49]. The high dimensionality leads to the so-called curse of dimensionality, creating a situation in which all clusters appear almost equally distant from one another. Despite these challenges, the embeddings are intended to undergo thorough testing in later evaluations. Additional tasks that leverage sector and industry classifications have been incorporated. Specifically, these tasks entail Concept Categorization and Outlier Detection, as well as the idea to cluster distance vectors.

Concept Categorization The concept underlying the categorization strategy delineated in [227] [6] posits that two word clusters, when categorized (whether semantically, syntactically, or on other bases), should consistently replicate the same categorical structures upon clustering their embeddings. When this principle is applied to the S2V Model, a stock c_i can be aligned with a specific category k, such as an industry or sector $(K/\dot{K})$.

Upon the application of a clustering algorithm, such as k-means, to these embeddings, the ideal outcome would be $\forall c_i, c_j : K[i] = K[j] \Rightarrow i, j \in U$ (with U being a K-means cluster). As indicated in the referenced literature ([227] [6]), it is not necessary for all $c \in C$ to be precisely categorized in the initial pass. Instead, pairwise comparisons between two categories (for example, comparing two different industries or sectors) can be conducted. Although a higher number of clusters is less typical in the literature, it remains a feasible approach and will be tested here.

The results of the respective binary cluster analysis and all clusters at once are shown in Table 7.10. A comparison of the markets for the [5] run has also been included. For the industries, the dataset D with $D \subset \dot{K} \bowtie \dot{K}$ with $|D| = 100$ and at least two stocks being in each sampled industry was used. Due to the concerns listed in Section 6.2 when using industries, the good performance here could be somewhat misleading. K-means was used for clustering, and each run was repeated 10 times. The purity metric from [152] is used as an evaluation metric.

Table 7.10: Comparison of the purity in the concept categorization task of different runs on different categories.

Approach	Sector		Industry		Country	
Evaluation	All	Binary	All	Binary	All	Binary
[1]	0.228	0.690	0.550	0.864		
[2]	0.193	0.682	0.560	0.881		
[3]	0.233	0.686	0.553	0.882		
[4]					0.418	0.772
[5]	0.264	**0.725**	0.559	0.878		
[6]	0.000	0.714	0.000	**0.958**		

Certain regressive company metrics, such as market capitalization, EBITDA, PERatio, cash, Volume, and debt, reflect a firm's status over extended periods while still having a temporal component. Yi et al. [255] have already proposed the prediction of country or company size using the embeddings. The exploratory analysis involved clustering based on these real-valued metrics to categorize stocks (e.g. clustering the ten most indebted c_i and of the ten least indebted c_i). Clustering purity was attempted to be enhanced by adapting the purity measure from [6], modifying it to penalize clusters with high deviations from the cluster mean. The adjusted purity formula, extending the original definition from [152]. Despite these adjustments, the approach did not yield significant insights, likely due to the coarse granularity of the metrics (EBITDA or EPS) and the insufficiency of the embeddings to capture nuances in financial attributes like debt or cash as they might can not be learned from daily returns. The clustering by real-numbered values proved ineffective in the experiments, also for the [2] run, and thus, was not pursued further.

Outlier Detection The outlier-detection method of Camacho-Collados and Navigli [11] is directly applicable to SF using sectors and industries. In their work, the model is tasked with identifying an incongruent element within a set of words, based on semantic, syntactic, or other disparities. The procedure involves the computation of what is termed the 'compactness score' for the corresponding embeddings, which quantifies the pairwise semantic similarity among the elements of the set. The element characterized by the minimal compactness score is posited as the outlier. The Outlier Position Percentage (OPP) score can be adapted from [11]. The results can be seen in Table 7.11. For [6] $|C|$ was to small for meaningful results.

Table 7.11: Comparison of the OOP of different runs on outlier detection on different categories.

Approach	Sector	Industry	Country
[1]	0.295	0.319	
[2]	0.283	0.245	
[3]	0.384	**0.441**	
[4]			0.374
[5]	0.203	0.431	

Some cross-sector clusters selected by hand with run [1] have also been checked. Here the results are sometimes much better. The finance sector is still particularly difficult, presumably because the embeddings are far out anyway.

Table 7.12: Comparison of the OOP of different runs on outlier detection on self-selected datasets.

Cluster	Outlier	OOP
SCHW, C, GS, MS, BAC	MSFT	0.66
MSFT, IBM, AMZN, GOOG, AAPL, NVDA, EA, GOOGL, EBAY	PEP	1
AMT, EQIX, MAR, PSA, CCI	C	0.8

TABLE 7.13: Comparison of the OOP of different runs on outlier detection on self-selected datasets for [6].

Cluster	Outlier	OOP
SCHW, C, HBAN, UNH	NKE	1
AAPL, ADSK, JNPR	PH	1

Additional outlier tests on self-selected data (Table 7.12, Table 7.13) were also unsuccessful.

Clustering Distance Vectors In the study, the author experimented with computing distance vectors $\mathbf{e}_i - \mathbf{e}_j$ and subsequently clustering these vectors using Mean Shift and DBSCAN [59] to identify potential recurrent patterns. This procedure was motivated by the observation discussed in Section 2.1, where Word2Vec embeddings often exhibit the phenomenon that semantically or syntactically related word pairs correspond to vectors with similar directions or magnitudes. Transferring this intuition to the financial domain, it was hypothesized that stock pairs might exhibit analogous relational patterns in the embedding space, despite the lack of fixed or explicitly defined relationships between them. To explore this possibility, distance vectors $\mathbf{e}i - \mathbf{e}j$ were computed and clustered, with the expectation that recurrent patterns—if present—could be detected in the form of groups of similar relation i.e. distance vectors. Such clusters would then have provided a basis for interpreting and categorizing latent inter-stock relationships. However, as described, the clustering algorithms did not reveal meaningful groupings, suggesting that this approach may not be effective in capturing structured relationships among stocks. To the best of the author's knowledge, this approach is not used in NLP, likely because relationships are predefined there. However, neither clustering algorithm revealed obvious clusters within the vector space, leading to the conclusion that this method may not be effective for uncovering semantic or syntactic patterns in this context.

7.2.4 Embedding-Analysis

The following presents PCA visualizations of different runs. PCA is a dimensionality reduction technique that transforms a set of possibly correlated variables into a smaller number of uncorrelated variables called principal components. These components are obtained by projecting the data onto directions of maximal variance, which are determined through an eigenvalue decomposition of the covariance matrix (or equivalently, via singular value decomposition). In practice, PCA is used to reduce redundancy and aid visualization while preserving variance. In this work, PCA is applied to map high-dimensional vector representations into two or three dimensions, enabling interpretable visualization of the latent structures.

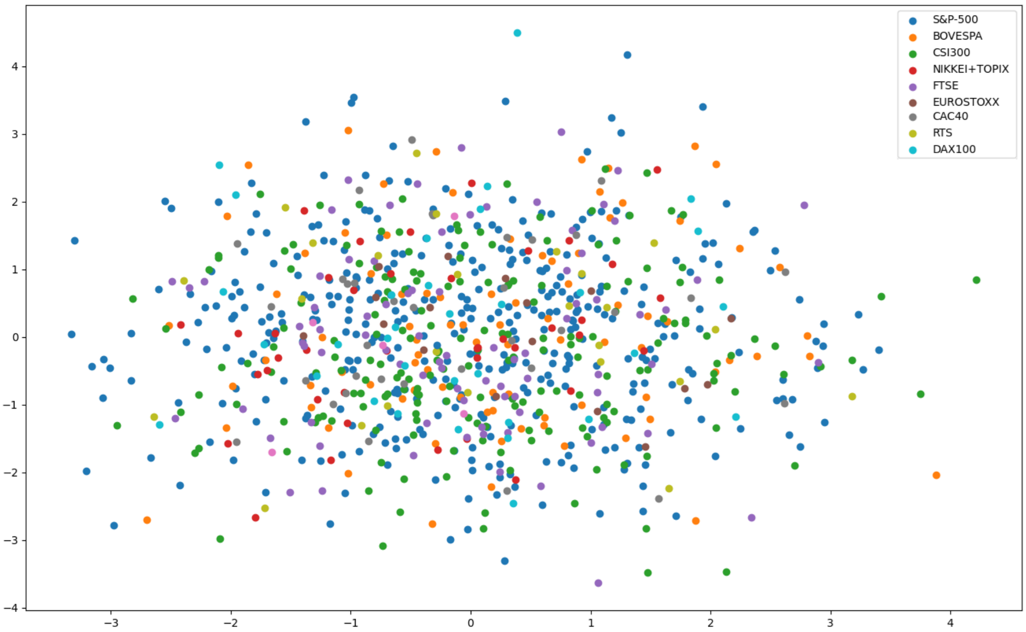


FIGURE 7.3: 2D PCA visualization of the embeddings of the [4] runs with coloring of the embeddings according to different markets.

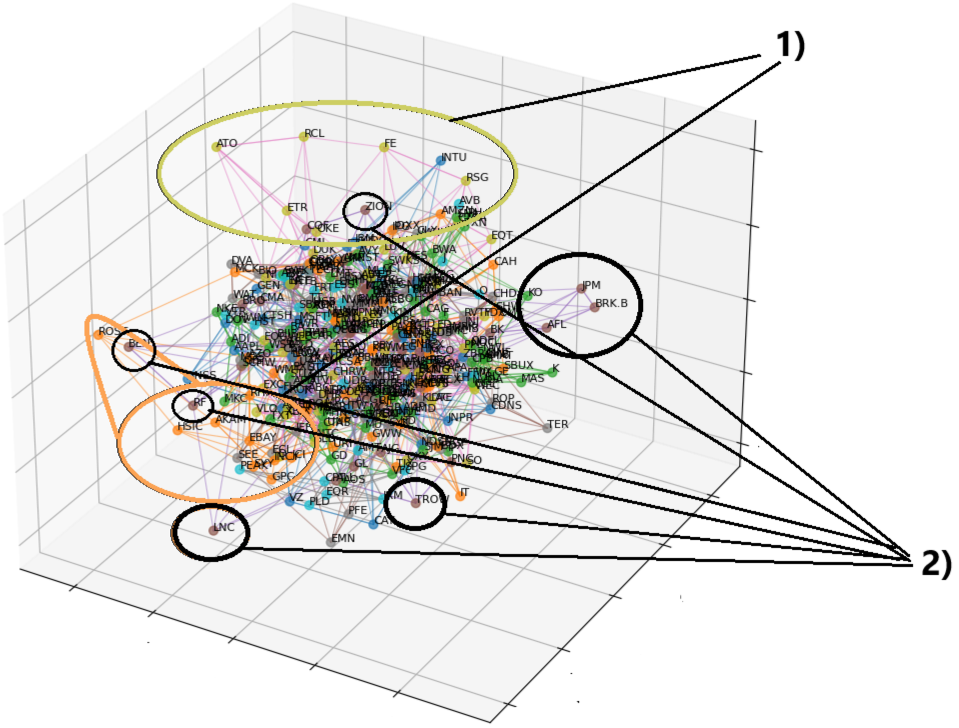


Figure 7.4: 3D PCA representation of the S2V embeddings of the [1] run. The sectors are shown in the respective colors and the five nearest neighbors are connected. The areas of the observations are marked.

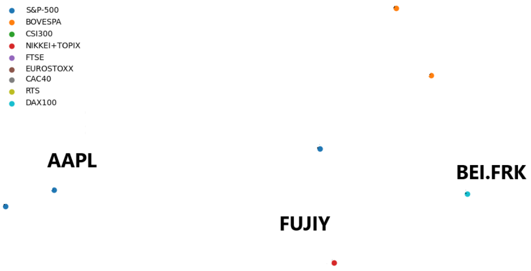


Figure 7.6: Section of 2D PCA representation of embeddings from [1] run. Observation 5) can also be seen here for international markets. The ticker names of the other stocks have been removed for presentation purposes.

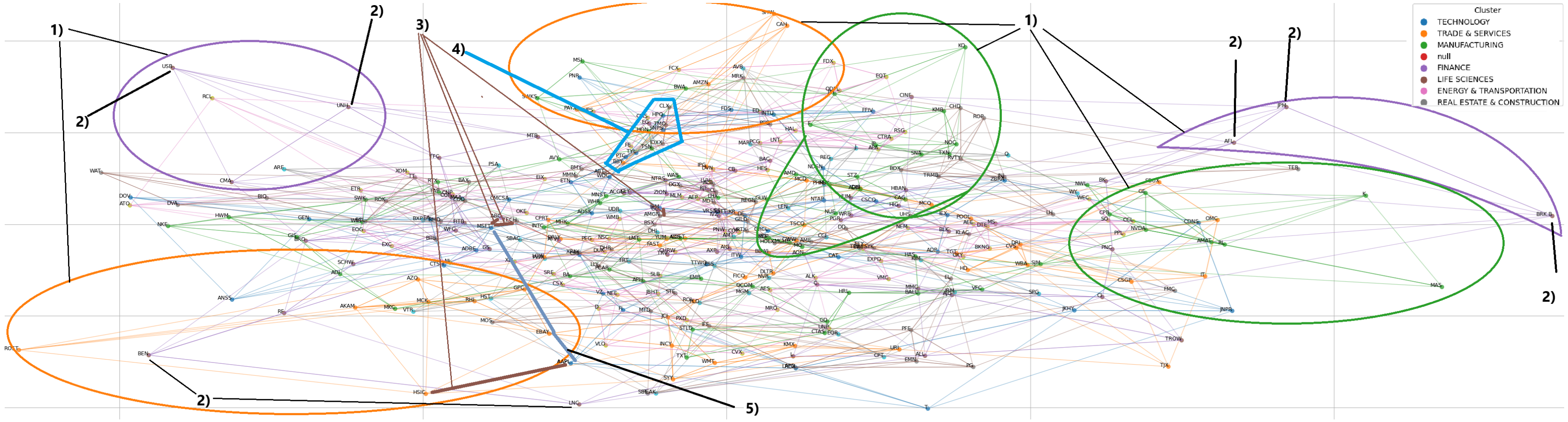


FIGURE 7.5: 2D PCA representation of the S2V embeddings of the [1] run. The sectors are shown in the respective colors and the five nearest neighbors are connected. The areas of the observations are marked.

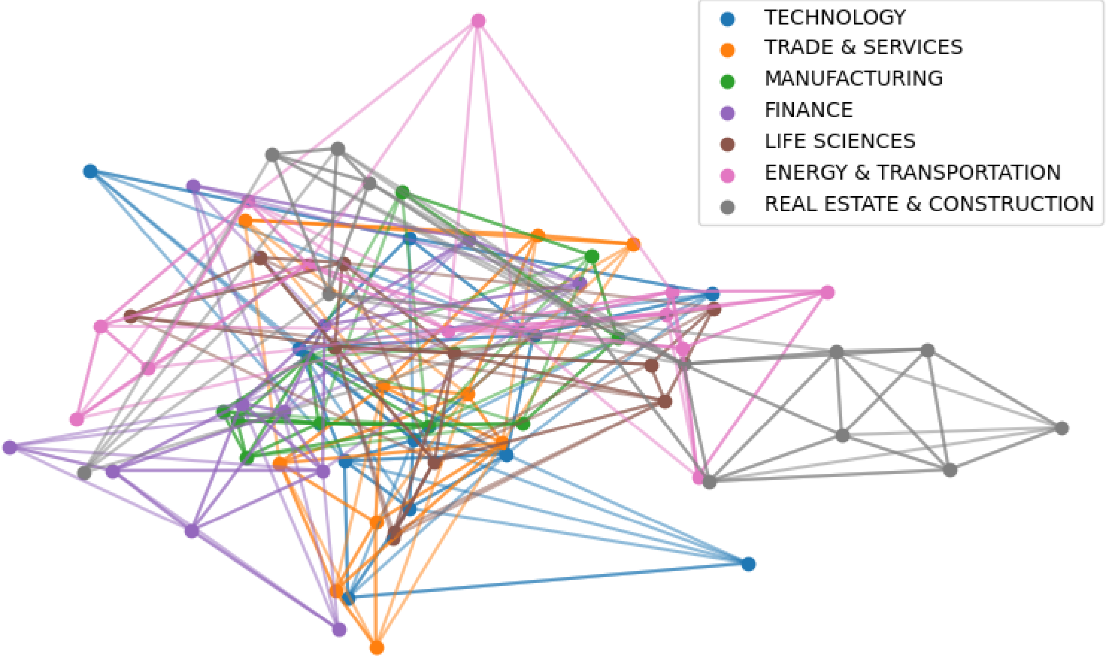


Figure 7.7: Section of 2D PCA representation of embeddings from [6] run i.e. the T5 based context sensitive embeddings. The ticker names of the other stocks have been removed for presentation purposes.

The following discussion presents various phenomena observed in the C-CBOS SMC runs, with a specific focus on the example as depicted in [1] for the S&P-500 🇺🇸 dataset. Despite the lack of clear clustering by industry, sector, or country in the S2V algorithms, as highlighted in the previous Section 7.2.3 and the high accuracy of the C-CBOS SMC models remains convincing. Notable patterns and clustering behaviors in the data are summarized. Generally, the (AV database) sector assignments for certain stocks are questionable. For instance, the computer hardware producer AMD 🏭 is classified within the manufacturing 🏭 cluster, and the online retailer Amazon AMZN 🛒 is categorized under trade & services 🛒. Other examples include UNH 💰 and AFL 💰, primarily known for their operations in life science and health insurance 🧬, yet they are grouped under the finance sector 💰.

In Figure 7.5 an overview of certain phenomena is marked which will be discussed in the following:

1) Sector Clusters: Although not distinctly segregated, certain cluster structures can be recognized for individual industries. Noteworthy are the two finance, two manufacturing clusters and the trade and services clusters, each showing internal coherence yet overlapping with adjacent clusters to some extent. This can also be seen in the 3D representation in Figure 7.4.

2) Finance Outliers: Companies in the finance sector, such as JPM 🪙, BEN 🪙, LNC 🪙, USB 🪙, UNH 🪙, BRK.B 🪙, and AFL 🪙, tend to exhibit higher distances from other stocks. Using BRK.B 🪙 as an example, its average correlation with other stocks is notably low at -0.383, compared to the average correlation across all S&P-500 🇺🇸 stocks of 0.44407 with a standard deviation of 0.3994. This indicates that BRK.B 🪙 operates atypically relative to the market. The underlying reasons for these observations may stem from the resilience of the mentioned stocks to broader economic fluctuations. Financial institutions such as JPM 🪙 are structured to generate profits across diverse market conditions. Additionally, holdings like BRK.B 🪙 have a big set of business sectors—from insurance, investment, and real estate to manufacturing, transportation, and natural gas utilities[2]. This diversification might enable them to operate with a degree of independence from typical market correlations, potentially insulating them from sector-specific downturns and enhancing their stability in volatile markets. In Figure 7.7 a finance cluster can be seen as well.

3) Sector Proximity Phenomenon: There is a noticeable proximity between electronics/tech companies and life science/bio research firms. Examples include the alignments of HSIC 🛒 with AAPL 💡, IBM 💡 with AMGN 🧬, and MSFT 💡 with TECH 🧬 and ABC 🧬. This pattern also extends to international markets, as seen in the [4] run represented in Figure 7.6, where AAPL 🇺🇸 and FUJIY 🇯🇵 are close to the life science company BEI.FRK 🇩🇪. Otherwise, As can be seen in Figure 7.3, there is no obvious structure across international markets.

[2] https://www.berkshirehathaway.com/subs/sublinks.html

4) Electronics and Semiconductor Cluster: There is a distinct cluster comprising semiconductor and electronics manufacturers, indicating a specific industry concentration within the dataset.

5) Tech Giants Cohesion: A broad cluster includes tech giants like AAPL 💡, MFST 💡, and EBAY 🛒, demonstrating close proximity amongst these major players. These observations provide a preliminary view of clustering and relational dynamics within the S&P-500 🇺🇸. This is a preliminary analysis highlighting phenomena for further study.

7.3 CWRNN Results

The results for the CWRNN are in Table 7.14 and Table 7.15.

Table 7.14: SPP results for CWRNN.

$\mathbb{P}$	Interval	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
[1, 2, 3, 4, 5]	Interday	1.665/1.398	1.667/1.400	0.493/0.519	0.038/0.034	0.565/0.594
[1, 2, 2, 1]	Interday	1.469/1.392	1.468/1.394	0.517/0.505	0.008/0.002	0.623/0.607
[1, 2, 3, 5, 10, 5, 3, 2, 1]	Interday	1.469/1.392	1.468/1.394	**0.520/0.507**	0.001/0.007	0.654/0.634
[1, 2, 2, 1]	60min	0.395/0.344	0.396/0.344	**0.505/0.504**	0.011/0.009	0.506/0.502
[1, 2, 3, 4, 5]	60min	0.395/0.344	0.396/0.344	0.503/0.504	0.007/0.008	0.497/0.498
[1, 2, 3, 5, 10, 5, 3, 2, 1]	60min	0.394/0.343	0.394/0.343	0.502/0.503	0.005/0.007	0.499/0.499
[1, 2, 2, 1]	1min	0.064/0.062	0.064/0.062	**0.502/0.503**	0.004/0.005	0.482/0.483
[1, 2, 3, 4, 5]	1min	0.064/0.062	0.064/0.062	0.502/0.503	0.004/0.005	0.482/0.482
[1, 2, 3, 5, 10, 5, 3, 2, 1]	1min	0.064/0.062	0.064/0.062	0.502/0.502	0.003/0.003	0.481/0.481

Table 7.15: SMP results for CWRNN.

$\mathbb{P}$	Interval	Acc ↑	MCC ↑	F1 ↑
[1, 2, 3, 5, 10, 5, 3, 2, 1]	Interday	0.502/0.500	0.003/-0.000	0.509/0.505
[1, 2, 2, 1]	60min	0.518/0.518	0.025/0.034	0.455/0.496
[1, 2, 3, 8, 10]	60min	0.520/0.523	0.025/0.033	0.427/0.447
[1, 2, 3, 5, 10, 5, 3, 2, 1]	60min	**0.516/0.527**	0.008/0.036	0.390/0.412
[1, 2, 2, 1]	1min	**0.508/0.508**	0.007/0.010	0.459/0.468
[1, 2, 3, 8, 10]	1min	0.503/0.503	0.006/0.005	0.481/0.477
[1, 2, 3, 5, 10, 5, 3, 2, 1]	1min	0.503/0.502	0.003/0.003	0.476/0.478

7.4 QMSE Results

This section is mainly based on the authors publication [223] with new results for additional measurements.

As previously elucidated in Section 7.2, the Doc2Vec adaption QMSE can be assessed through two distinct methodologies. The evaluation of the regressive SPE task has been described in [223]; details can be found there. The SPE results can be seen in Table 7.17. For the additional datasets in this thesis and the SMC inspired reconstructions, the results have been listed in Section 7.4.

In the authors study in [223], various configurations of ρ and the layer dimensions were explored. Additional experiments were conducted with models distinct from the previously utilized simple neural networks, incorporating architectures such as RNNs, LSTMs, and transformers. The findings suggest that models of increased complexity enable more effective reconstruction of inputs, attributable to their enhanced capacity for information encoding. Conversely, increased model complexity has been observed to reduce abstraction and degrade the quality of the encoded representation $\mathbf{e}$. This approach is set apart from the multi-layer architectures commonly used in related work such as [5].

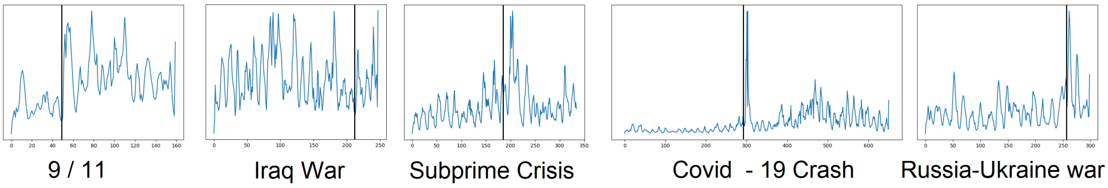
The QMSEs produced by $F^{\langle \mathrm{A} \rangle}$ effectively capture complex market dynamics. This effect has been observed in the identification of infrequent market anomalies, such as financial downturns, using $\mathbf{e}$ or d. However, models using transformer architectures have not consistently produced stable results, as shown by the large variation in loss during unreported training sessions, which is especially noticeable at lower values of ξ. Remarkably, the pairwise mean distances among embeddings produced by transformer models, quantified at 1.41 when $\xi = 64$, are significantly lower compared to the mean distance of 2.46 characteristic of $F^{\langle \mathrm{A} \rangle}$-derived embeddings. Moreover, the application of clustering algorithms like DBSCAN [59] to organize transformer-generated QMSEs into coherent structures has met with limited success. Notably, DBSCAN frequently fails to delineate distinct clusters, often relegating all vectors $\mathbf{e}$ to the category of noise. Although RNN-based models produce relatively stable loss values, the issues of small distances and the absence of clear clustering structures remain unresolved. Despite strong reconstruction performance, as measured by sMAPE and accuracy, the learned embeddings do not fully capture the complexity of market dynamics. During the author's investigation, it was found that a detailed analysis of each market scenario must be placed in a temporal context using κ or Δt. Employing a flattening function $f(.)$ was essential for learning robust embeddings.

Table 7.16: QMSE SMC results.

C	Feature	dim(e)	Acc ↑	MCC ↑	F1 ↑
S&P 500 🇺🇸	OHCLV	64	0.667/0.682	0.330/0.364	0.635/0.673
S&P 500 🇺🇸	OHCLV	256	0.679/0.694	0.355/0.387	0.649/0.686
S&P 500 🇺🇸	OHCLV	1024	0.659/0.672	0.314/0.343	0.627/0.663
S&P 500 🇺🇸	Close	64	0.694/0.720	0.384/0.440	0.659/0.711
S&P 500 🇺🇸	Close	256	**0.718/0.742**	0.432/0.484	0.687/0.734
S&P 500 🇺🇸	Close	1024	0.694/0.718	0.382/0.436	0.658/0.710
S&P 500 🇺🇸	Volume	64	0.653/0.641	0.305/0.280	0.673/0.665
S&P 500 🇺🇸	Volume	256	0.669/0.659	0.338/0.317	0.687/0.679
S&P 500 🇺🇸	Volume	1024	0.691/0.682	0.381/0.363	0.705/0.698
All$^{(2010:)}$	OHCLV	64	0.625/0.644	0.250/0.287	0.609/0.629
All$^{(2010:)}$	OHCLV	256	0.598/0.619	0.196/0.236	0.579/0.598
All$^{(2010:)}$	OHCLV	1024	0.637/0.654	0.274/0.307	0.625/0.640
All$^{(2010:)}$	Close	64	0.671/0.690	0.342/0.378	0.648/0.673
All$^{(2010:)}$	Close	256	0.696/0.709	0.392/0.417	0.679/0.695
All$^{(2010:)}$	Close	1024	**0.705/0.716**	0.409/0.432	0.688/0.704
All$^{(2010:)}$	Volume	64	0.610/0.610	0.218/0.220	0.629/0.628
All$^{(2010:)}$	Volume	256	0.614/0.615	0.226/0.229	0.631/0.631
All$^{(2010:)}$	Volume	1024	0.610/0.613	0.220/0.225	0.628/0.629

Table 7.17: QMSE SPE results.

C	Feature	dim(e)	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
S&P 500 🇺🇸	OHCLV	64	0.006/0.005	0.006/0.005	0.555/0.553	0.108/0.106	0.574/0.567
S&P 500 🇺🇸	OHCLV	256	0.006/0.005	0.006/0.005	0.638/0.647	0.274/0.295	0.656/0.652
S&P 500 🇺🇸	OHCLV	1024	0.006/0.005	0.006/0.005	0.637/0.643	0.273/0.285	0.652/0.646
S&P 500 🇺🇸	Close	64	0.002/0.002	0.002/0.002	0.705/0.725	0.408/0.450	0.726/0.728
S&P 500 🇺🇸	Close	256	0.002/0.002	0.002/0.002	0.712/0.733	0.422/0.467	0.733/0.738
S&P 500 🇺🇸	Close	1024	0.002/0.002	0.002/0.002	**0.731/0.749**	0.459/0.499	0.748/0.753
S&P 500 🇺🇸	Volume	64	0.017/0.017	0.017/0.017	0.628/0.613	0.257/0.227	0.625/0.610
S&P 500 🇺🇸	Volume	256	0.017/0.017	0.017/0.017	0.624/0.609	0.249/0.219	0.623/0.609
S&P 500 🇺🇸	Volume	1024	0.017/0.016	0.017/0.016	0.637/0.624	0.275/0.248	0.635/0.619
All$^{(2010:)}$	OHCLV	64	0.018/0.020	0.018/0.020	0.508/0.512	0.018/0.027	0.522/0.528
All$^{(2010:)}$	OHCLV	256	0.020/0.021	0.020/0.021	0.528/0.539	0.056/0.078	0.527/0.537
All$^{(2010:)}$	OHCLV	1024	0.028/0.027	0.028/0.027	0.524/0.530	0.048/0.060	0.520/0.526
All$^{(2010:)}$	Close	64	0.004/0.004	0.004/0.004	0.513/0.519	0.027/0.038	0.518/0.524
All$^{(2010:)}$	Close	256	0.003/0.003	0.003/0.003	0.612/0.634	0.225/0.268	0.615/0.636
All$^{(2010:)}$	Close	1024	0.003/0.002	0.003/0.002	**0.688/0.702**	0.376/0.405	0.689/0.701
All$^{(2010:)}$	Volume	64	0.072/0.069	0.071/0.069	0.523/0.535	0.049/0.073	0.527/0.538
All$^{(2010:)}$	Volume	256	0.071/0.070	0.070/0.070	0.512/0.522	0.027/0.047	0.516/0.524
All$^{(2010:)}$	Volume	1024	0.076/0.079	0.075/0.078	0.512/0.518	0.027/0.039	0.519/0.525



9 / 11 Iraq War Subprime Crisis Covid - 19 Crash Russia-Ukraine war

Figure 7.8: Temporal variation of d during critical economic incidents over the past twenty-four years. The vertical axis, depicting d, remains unlabeled to accommodate the considerable fluctuation in values across different events, favoring a relative over an absolute representation for enhanced clarity in visualization. The figure is taken from [223].

Regarding the intrinsic evaluation, the findings from [223] are briefly mentioned as outlined in Section 7.4.

Intrinsic Evaluation As an initial check, the ability of QMSE embeddings to identify anomalies in stock price trajectories was evaluated. This examination focused on five significant market downturns subsequent to the year 2000. The embeddings $\mathbf{e}$, corresponding to periods immediately preceding and following each downturn, are delineated in Figure 7.14. Additionally, the distances d between

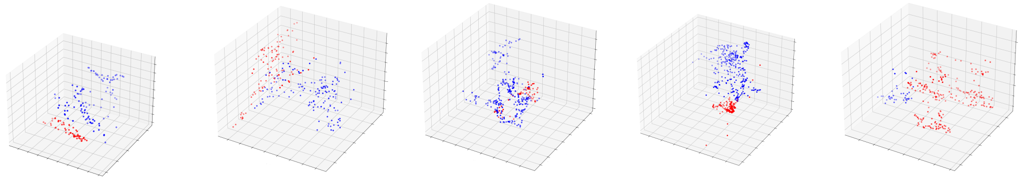


Figure 7.9: 9/11, 2001

Figure 7.10: Iraq War, 2003

Figure 7.11: Subprime Crash, 2008

Figure 7.12: Covid-19, 2020

Figure 7.13: Ukraine War, 2022

Figure 7.14: 3D PCA visualization illustrating the trajectory of **e** through five significant events that induced notable stock market volatility. Red markers represent the state of **e** prior to each event, whereas blue markers denote the dynamics of **e** following the occurrence of these events. The figure is taken from [223].

consecutive embeddings are depicted in Figure 7.8. Apart from a few exceptions, these downturns can be seen clearly in the figures.

The graphical depictions associated with the Subprime Crisis and the market downturn induced by the Iraq war demonstrate a comparatively reduced level of clarity relative to other events. For the Subprime Crisis, the lower clarity may result from events leading up to the Lehman Brothers bankruptcy, which acted as a trigger rather than a single event. Similarly, market fluctuations during the Iraq war may be linked to rising political tensions before the conflict [161].

The next intrinsic method considered in [223] was the NNA, which can be interpreted as an absolute scoring method. However, this interpretation relies on the assumption that similar market situations result in similar future developments—a premise that many economists are likely to question. Similar findings are presented in Section 7.7.4.

This method, discussed in [223], is used to evaluate intrinsic properties of the embeddings. While the NNA generally performs poorly for most price features (e.g. Close Price, Open Price etc.), it yields relatively strong results for trading Volume. Specifically, the NNA achieves accuracy values of up to 0.63 ± 0.03 and 0.56 ± 0.02 for the inverted distance approach.

7.5 $F^{\langle \mathrm{T} \rangle}$ Results

The results for experiments on the $F^{\langle \mathrm{T} \rangle}$ models are listed in the following. If not indicated differently $\Delta t = 128$ is set for all experiments in this section. For all models, $\theta_\kappa = 32$ was chosen.

7.5.1 Pre-Training

An overview of pretraining results for $F^{\langle \mathrm{T} \rangle}$-based architectures is presented below. For clarity, runs of recurrent transformer models with suboptimal performance were excluded.

Pre-training with $\rho_{\text{heads}} = 1$ For the approach where $\rho_{\text{heads}} = 1$, the corresponding results for the SMC method are presented in Table 7.18 and Table 7.19. The approach with $\rho_{\text{heads}} = 1$ poses several challenges. Firstly, it requires substantial computational resources to compute the softmax over $(\theta_\kappa)^2$ vectors. Secondly, the advantages of multi-head attention are eliminated. Most importantly, significantly worse SMP/SPP performance results were achieved.

One advantage of this configuration is its improved performance during pretraining, which can only be matched by $F^{\langle \mathrm{L} \rangle}$ in the $F^{\langle \mathrm{J\text{-}M} \rangle}$ model. Selected intraday results are provided below for completeness. Due to limited performance variation—such as the similar 60min MFM SMC results between $F^{\langle \mathrm{BM\text{-}T} \rangle}$ and $F^{\langle \mathrm{E\text{-}M} \rangle}$—and resource constraints, the intraday analysis is restricted to a few example results. The additional results for SMC are summarized in Table 7.22, and those for SPE can be found in Table 7.23. Due to resource limits, intraday runs were not trained for the same duration as interday runs, which might explain the performance differences. The multi-context model in Table 7.23 can be taken as an example of how the $F^{\langle \mathrm{N} \rangle}$ and the additional contexts often only confuse the model and lead to poorer performance.

Table 7.19: Results for SMC - MFM Interday runs with $\rho_{\text{heads}} = 1$.

Model	Acc ↑	F1 ↑	MCC ↑
Baseline	**0.701/0.697**	0.707/0.704	0.402/0.394
E-M	0.549/0.545	0.605/0.607	0.100/0.091
J-M	0.693/0.685	0.694/0.692	0.387/0.369
E-M-N	0.660/0.653	0.673/0.671	0.322/0.305
J-C	0.672/0.672	0.667/0.673	0.345/0.345
L-M	0.708/0.699	0.709/0.704	0.416/0.398
L-C-N	0.510/0.509	0.554/0.562	0.020/0.016

Table 7.18: Results for SMC - MPM Interday runs with $\rho_{\text{heads}} = 1$.

Model	Acc ↑	F1 ↑	MCC ↑
Baseline	0.672/0.671	0.669/0.674	0.346/0.343
E-M	0.673/0.660	0.669/0.663	0.347/0.319
J-M	0.523/0.522	0.504/0.567	0.046/0.043
E-M-N	0.656/0.650	0.653/0.658	0.312/0.300
J-M-N	0.661/0.654	0.650/0.651	0.323/0.308
J-C	**0.680/0.677**	0.678/0.685	0.360/0.354
J-C-N	0.507/0.513	0.612/0.620	0.015/0.021
L-C	0.658/0.655	0.649/0.655	0.316/0.311
L-M-N	0.676/0.675	0.668/0.673	0.352/0.349
L-C-N	0.510/0.511	0.553/0.567	0.021/0.019

Table 7.20: Results for SPE - MPM Interday runs with $\rho_{\text{heads}} = 1$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
Baseline	0.418/0.408	0.419/0.408	**0.674/0.671**	0.670/0.678	0.349/0.342
E-M	0.419/0.409	0.420/0.408	0.672/0.670	0.670/0.676	0.347/0.341
J-M	0.467/0.447	0.467/0.447	0.594/0.574	0.600/0.583	0.188/0.147
E-M-N	0.425/0.409	0.425/0.409	0.676/0.669	0.672/0.664	0.352/0.338
J-M-N	0.423/0.406	0.423/0.406	0.668/0.665	0.672/0.680	0.336/0.331
J-C	0.420/0.406	0.420/0.406	0.676/0.670	0.675/0.681	0.352/0.340
L-M	0.420/0.406	0.420/0.406	0.675/0.667	0.670/0.672	0.351/0.334
L-C	0.421/0.406	0.421/0.406	0.674/0.666	0.676/0.680	0.348/0.331
L-M-N	0.426/0.409	0.426/0.409	0.673/0.668	0.658/0.667	0.347/0.337
L-C-N	0.420/0.405	0.420/0.405	0.679/0.668	0.673/0.674	0.357/0.336

Table 7.21: Results for SPE - MFM Interday runs with $\rho_{\text{heads}} = 1$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
Baseline	0.412/0.402	0.412/0.402	0.676/0.668	0.685/0.681	0.351/0.334
E-M	0.451/0.431	0.451/0.431	0.547/0.533	0.553/0.546	0.094/0.065
J-M	0.447/0.429	0.447/0.429	0.583/0.579	0.636/0.632	0.173/0.160
E-M-N	0.451/0.432	0.451/0.432	0.558/0.538	0.544/0.534	0.116/0.076
J-M-N	0.424/0.410	0.424/0.410	0.660/0.648	0.668/0.666	0.320/0.296
J-C	0.452/0.432	0.452/0.432	0.546/0.532	0.574/0.565	0.092/0.063
J-C-N	0.421/0.409	0.421/0.409	0.669/0.663	0.666/0.668	0.338/0.325
L-C	0.422/0.407	0.422/0.407	0.662/0.664	0.669/0.678	0.324/0.328
L-M-N	0.423/0.408	0.423/0.408	**0.672/0.669**	0.665/0.674	0.344/0.337

Table 7.22: Summary of SMC results for MPM and MFM at 60min interval with $\rho_{\text{heads}} = 1$.

Model	Task	Acc ↑	F1 ↑	MCC ↑
Baseline	MFM	0.532/0.533	0.480/0.477	0.067/0.066
E-M	MFM	**0.533/0.534**	0.473/0.471	0.068/0.067
E-M-N	MFM	0.507/0.506	0.484/0.486	0.012/0.010
J-M $+F^{\langle \mathrm{R} \rangle}$	MFM	0.508/0.506	0.494/0.496	0.016/0.012
Baseline	MPM	**0.532/0.533**	0.478/0.475	0.065/0.065

Table 7.23: Summary of SPE results for MPM and MFM at 60min and 1min intervals with $\rho_{\text{heads}} = 1$.

Model	Task	Interval	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
Baseline	MFM	60min	0.177/0.171	0.177/0.171	**0.598/0.595**	0.197/0.191	0.613/0.609
Baseline	MFM	1min	0.131/0.131	0.131/0.131	0.531/0.525	0.522/0.514	0.062/0.051
Baseline	MPM	60min	0.179/0.173	0.179/0.173	**0.593/0.589**	0.606/0.600	0.187/0.180
E-M	MPM	60min	0.179/0.174	0.179/0.174	0.583/0.579	0.592/0.586	0.166/0.160
Baseline	MPM	1min	0.131/0.130	0.131/0.130	**0.532/0.527**	0.521/0.514	0.064/0.053

Pre-training with $F^{\langle \mathbf{L} \rangle}$ For the interday runs employing the $F^{\langle \mathrm{L} \rangle}$ approach, stable and satisfactory results were often not obtained, with exception of the MFM $F^{\langle \text{J-M} \rangle}$ configuration. The corresponding results for this setup are presented in Table 7.24. All other MFM runs yielded suboptimal performance, with accuracy levels falling below 53.5% and MPM accuracy remaining below 52.0%.

Table 7.24: Results for SMC - MFM Interday runs with $F^{\langle \mathrm{L} \rangle}$.

Model	Acc ↑	F1 ↑	MCC ↑
J-M	0.673/0.654	0.671/0.659	0.346/0.308

Table 7.25: Results for SMC - MPM 60min runs with $F^{\langle \mathrm{L} \rangle}$.

Model	Acc ↑	F1 ↑	MCC ↑
E-M	0.571/0.557	0.507/0.501	0.137/0.111
J-M	**0.619/0.615**	0.588/0.595	0.236/0.229
E-M-N	0.611/0.607	0.559/0.577	0.219/0.213

Table 7.26: Results for SMC - MFM 60min runs with $F^{\langle \mathrm{L} \rangle}$.

Model	Acc ↑	F1 ↑	MCC ↑
E-M	0.560/0.553	0.481/0.478	0.114/0.103
J-M	0.534/0.534	0.475/0.472	0.069/0.067
J-M	0.568/0.555	0.501/0.496	0.131/0.107
E-M-N	**0.576/0.559**	0.507/0.504	0.148/0.115

Table 7.27: Results for SMC - MPM 1min runs with $F^{\langle \mathrm{L} \rangle}$.

Model	Acc ↑	F1 ↑	MCC ↑
Baseline	0.545/0.544	0.277/0.265	0.081/0.071

Table 7.28: Results for SPE - MPM Interday runs with $F^{\langle \mathrm{L} \rangle}$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
E-M	0.453/0.428	0.454/0.428	**0.571/0.587**	0.567/0.593	0.141/0.174
J-C	0.462/0.439	0.462/0.439	0.504/0.509	0.603/0.609	0.009/0.013

Table 7.29: Results for SPE - MPM 60min runs with $F^{\langle \mathrm{L} \rangle}$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
Baseline	0.192/0.187	0.192/0.187	0.533/0.524	0.565/0.555	0.067/0.050
E-M	0.316/0.314	0.315/0.314	**0.589/0.587**	0.585/0.585	0.178/0.175
L-C	0.197/0.192	0.197/0.192	0.510/0.508	0.555/0.551	0.021/0.019

The larger models like the $F^{\langle\text{L-C}\rangle}$ model in Table 7.29 converge after a few epochs.

Table 7.30: Results for SPE - MFM 60min runs with $F^{\langle\text{L}\rangle}$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
Baseline	0.419/0.412	0.419/0.411	0.505/0.504	0.532/0.527	0.010/0.010
J-M	0.252/0.249	0.252/0.249	**0.521/0.515**	0.583/0.583	0.045/0.036

In general, MPM was more difficult for the model than MFM, likely because information from other price features and step indicators is needed for accurate classification/estimation. The is supported by the results for $F^{\langle\text{X-CBOS}\rangle}$ in Section 7.2. This also holds for ASMs, which implicitly perform MPM and show stronger results. In Figure 7.15 a performance comparison in SMC of both masking approaches is shown as an example, which also shows that there is a performance 'jump' in MFM which can also be noticed it in the ASMs, but never in $F^{\langle\text{T}\rangle}$ based MFM. Also it can be seen how both approaches initially have a similar performance until the model learns to use the other features of the same stock and time step to achieve better MFM performance.

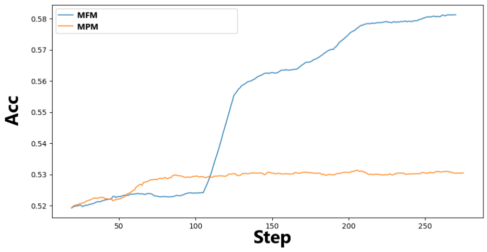


Figure 7.15: MPM vs. MPM visualization.

To analyze the transformer's behavior on masking tasks, relevance was visualized in Figure 7.16 using the approaches in Appendix A.9. The author has changed the proportion of masked features so that it is different for the time steps and it can be seen that the relevance increases with the number of masked prices in the time step (for this $\forall i, j : M[i, j] \sim \mathcal{B}(\frac{j}{\Delta t} \cdot \nu_{\text{MFM}})$ holds). The height indicates the number of masked features in the time step and the heatmap the relevance.

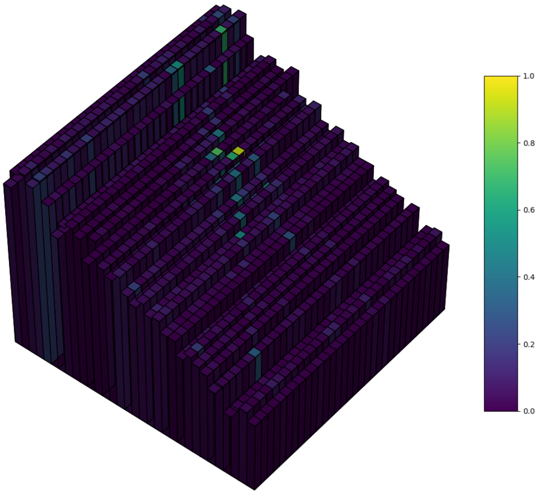


FIGURE 7.16: Relevance visualization for MFM.

Further insights derived from the masking tasks indicate that these tasks are generally easier for the models to generalize compared to trend prediction. However, performance on the validation and test sets remained poor. This is in contrast when compared with the ASMs discussed in the subsequent chapter. Both approaches utilize the same underlying information (with the exception of $|C|$), differing only in their representational format.

7.5.2 Finetuning

The results for TP are listed below.

Δt Comparison In all other experiments, Δt was tuned as a hyperparameter. To verify the thesis that the increased context length has a positive effect on performance, as indicated in [224], the best performance of the $F^{\langle \text{J-M} \rangle}$ has been listed in Table 7.31 and Table 7.32 for SMP and SPP respectively. Due to the enormous computing time, 1min runs were not possible and the 60min runs were only repeated twice.

TABLE 7.31: Results for SMP for different Δt.

Interval	Δt	Acc ↑	F1 ↑	MCC ↑
Interday	64	**0.503/0.502**	0.440/0.433	0.008/0.000
Interday	196	0.512/0.495	0.355/0.393	-0.004/0.003
60min	64	**0.515/0.510**	0.335/0.331	0.005/0.006
60min	196	0.516/0.478	0.332/0.361	0.002/0.015
60min	256	0.515/0.477	0.331/0.360	0.002/0.015

Table 7.32: Results for SPP for different Δt.

Interval	Δt	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
Interday	64	1.381/1.141	5.884/5.466	**0.534/0.502**	0.534/0.500	0.079/0.003
Interday	196	1.382/1.142	5.885/5.467	0.532/0.502	0.533/0.500	0.079/0.003
60min	196	0.293/0.266	0.782/0.254	**0.571/0.521**	0.466/0.469	0.108/0.037
60min	256	0.298/0.268	0.665/0.263	0.567/0.505	0.469/0.438	0.097/0.004

Without Additional Pre-Training The following presents the results of the runs without additional pretraining. Table 7.33 displays the outcomes of the SMP interday runs, Table 7.34 the SMP 60min runs, Table 7.35 the SMP 1min runs, Table 7.36 the SPP interday runs, Table 7.37 the SPP 60min runs, and Table 7.38 the SPP 1min runs. Models yielding particularly strong F1-scores in the SMP interday setting include $F^{\langle \text{J–C} \rangle}$ and $F^{\langle \text{L–M} \rangle}$ (cf. Table 7.33); in the SMP 60min setting, notable models are $F^{\langle \text{J–C} \rangle} + F^{\langle \text{R} \rangle}$ and $F^{\langle \text{L–C} \rangle}$ (cf. Table 7.34); and in the SPP Interday setting, all models perform comparably well (cf. Table 7.36). Remarkably, $F^{\langle \text{J–M} \rangle}$ fails to outperform the baseline in the SMP interday setting despite pretraining, resulting in a particularly poor performance.

Table 7.33: SMP Interday runs.

Model	Acc ↑	F1 ↑	MCC ↑
E-M	0.508/0.503	0.424/0.475	0.006/0.009
J-M	0.510/0.493	0.401/0.439	0.007/-0.009
E-M-N	0.504/0.504	0.302/0.313	0.003/0.005
J-C	**0.503/0.506**	0.505/0.517	0.006/0.010
L-M	0.505/0.503	0.504/0.512	0.009/0.004
L-C	0.501/0.499	0.491/0.500	0.002/-0.002

Table 7.34: SMP 60min runs.

Model	Acc ↑	F1 ↑	MCC ↑
E-M	0.514/0.512	0.329/0.323	0.001/0.002
J-M	**0.520/0.514**	0.267/0.259	0.005/0.002
J-C $+F^{\langle \text{R} \rangle}$	0.507/0.505	0.416/0.412	0.003/0.001
L-C	0.511/0.506	0.407/0.400	0.008/0.003

Table 7.35: SMP 1min runs.

Model	Acc ↑	F1 ↑	MCC ↑
J-M	0.517/0.516	0.319/0.312	0.003/0.004
E-M	**0.518/0.516**	0.306/0.304	0.004/0.004

Table 7.36: SPP Interday runs.

Model	sMAPE ↓	MAPE	Acc ↑	F1 ↑	MCC ↑
E-M $+\mathcal{L}_p$	1.381/1.143	5.919/5.489	0.530/0.502	0.533/0.501	0.078/0.003
J-M $+\mathcal{L}_p$	1.381/1.144	5.894/5.502	**0.531/0.503**	0.534/0.503	0.080/0.007
J-C	1.381/1.141	5.884/5.466	0.531/0.501	0.534/0.500	0.080/0.003

Table 7.37: SPP 60min runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
E-M	0.370/0.373	1.125/0.965	0.545/0.522	0.477/0.477	0.075/0.038
J-M $+\mathcal{L}_p$	0.352/0.354	1.013/0.885	**0.548/0.523**	0.482/0.477	0.082/0.042

Table 7.38: SPP 1min runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
J-M	0.064/0.056	0.017/0.008	0.527/0.526	0.502/0.501	0.052/0.050

Pre-trained on Masking Tasks The results for $F^{\langle \mathrm{T} \rangle}$ based models pretrained on masking tasks can be seen in Tables 7.39 to 7.44. All models are pretrained on MPM.

Table 7.39: SMP Interday runs pretrained.

Model	Acc ↑	F1 ↑	MCC ↑
E-M $+\mathcal{L}_p$+$F^{\langle \mathrm{R} \rangle}$	**0.501/0.503**	0.454/0.453	0.003/0.004
J-M	0.507/0.502	0.334/0.363	0.003/-0.017

Table 7.40: SMP 60min runs pretrained.

Model	Acc ↑	F1 ↑	MCC ↑
E-M $+\mathcal{L}_p$	0.512/0.514	0.383/0.374	0.005/0.006

Table 7.41: SMP 1min runs pretrained.

Model	Acc ↑	F1 ↑	MCC ↑
J-M $+\mathcal{L}_p$+$F^{\langle \mathrm{R} \rangle}$	0.516/0.515	0.344/0.334	0.005/0.005

Table 7.42: SPP Interday runs pretrained.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
E-M $+\mathcal{L}_p$	1.613/1.157	6.298/5.403	0.504/0.500	0.508/0.503	0.018/0.001
J-M $+\mathcal{L}_p$+$F^{\langle \mathrm{R} \rangle}$	1.619/1.162	5.862/5.569	**0.507/0.506**	0.504/0.504	0.029/0.010

Table 7.43: SPP 60min runs pretrained.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
E-M	0.347/0.360	1.024/0.942	0.548/0.522	0.480/0.479	0.078/0.040

Table 7.44: SPP 1min runs pretrained.

Model	sMAPE ↓	MAPE ↓	Acc ↑	F1 ↑	MCC ↑
J-M	0.065/0.056	0.017/0.008	0.527/0.527	0.501/0.501	0.051/0.051

Further Masking Tasks The results for MTM, MSM, and PMM are not presented in tabular form. For all these tasks, the models' performance proved to be extremely poor, especially when compared to MFM and MPM. Consequently, they are unsuitable as pretraining tasks, and no fine-tuning was performed on models that had been pretrained using these approaches. The worst performance was observed with TSM.

For completeness, experiments with a vocabulary-based masking task (Section 6.6) were conducted for MPM and MFM. However, this approach was ultimately discarded, as the model exhibited significant instability when low-dimensional vectors or even scalar values were mapped to the vocabulary. Furthermore, an alternative strategy was explored that involved merging S2V representations with timestep representations, followed by the application of a single masking operation per timestep. This was intended to provide the model with explicit information which stock was masked. In the end, the approach delivered no clear gains, is was not pursued further.

Another unsuccessful masking task — or attempt to improve performance on MTM, MSM, or PMM — involved predicting regression values by redefining $\hat{M}[i,j] = M \otimes \ddot{M}$ with $\ddot{M}[i,j] \sim \mathcal{B}(-1,1) \cdot \alpha$. This approach also failed to yield meaningful improvements, as the model was unable to achieve stable performance.

Trend Matching As previously observed in [224] and discussed in Section 6.9.2, TM also failed when applied to $F^{\langle \mathrm{T} \rangle}$ architectures. Because TM is a novel contrastive task, it remains unclear whether the poor performance is due to the task or the architecture. However, based on the number of conducted experiments, the former explanation is preferred. For completeness, an overview of the experiments and architectural modifications is provided below:

Initially, it is advisable to use the mean or a dedicated TM token instead of feeding all flattened tensor elements into a linear layer, as proposed in [263]. This approach helps mitigate instability during backpropagation, which otherwise becomes challenging due to the binary single-class decision applied to the input of size $\xi \cdot \Delta t$.

Furthermore, training requires a very low learning rate to ensure stability. Several stabilization techniques were additionally experimented with, including the introduction of extra layers and batch normalization, L2 regularization in the TM layer, focal loss, entropy regularization (using entropy instead of BCE), and hinge loss.

Test changes were also made to the architecture based on considering half of X before and after the TM token separately, after processing by $F^{\langle \mathrm{T} \rangle}$. Specifically, each half was put into a distinct LSTM, GRU, or RNN and then either summed up the last hidden state and used the average as a decision or set the target variable to $Y \in \{0,1\}^{\xi \times \Delta t}$ and $Y[i,j] = y$ and entered H from $F^{\langle \mathrm{T} \rangle}$ or the outputs of the LSTMs/RNNs/GRUs concatenated together into the loss function while the model decision was measured by voting of the output tensor fields.

The last experiment was to use a QMSE (or LSTM) based AE to learn a representation $\mathbf{e}_1/\mathbf{e}_2$ for the left and right halves of the input which was reconstructed by the AE and dense representation of the input and then given as $\mathbf{e}_1 \otimes \mathbf{e}_2$ into the linear TM layer. A similar approach was tried for seasonalities and trends in [280], albeit for completely different tasks. In this thesis it was done to prevent the model from making the same predictions for all inputs because it was forced by the autoencoder to learn representations with enough unique features. All experiments were unsuccessful; future work may clarify whether this can serve as a pretraining task.

Integrating S2V Models An alternative method investigated involved integrating S2V models by stacking scaled S2V representations, thereby decreasing the embedding dimension of ξ_{S2V}. To prevent excessive model complexity, PCA transformations were employed, and the S2V embedding was excluded from the backpropagation process. This strategy was not effective and did not yield satisfactory results.

7.6 ASM Embedding Based Approach

In the following, the results for the embedding-based LLM adaptations are discussed. In general, GPT-2 or T5 were unable to be successfully optimized for any of the evaluated approaches or datasets. Although these models sometimes performed slightly better than the baseline, the high standard deviations made the results unreliable. Consequently, their results are not reported in tabular form, and further investigation of these models is refrained from.

For interday data, embedding-based approaches proved largely ineffective. Neither BERT, TransformerXL, nor LLaMA could be successfully optimized for the SMP task. For the SPP task, only LLaMA demonstrated consistent optimization capability. The corresponding average performance metrics are presented in Table 7.45.

Table 7.45: SPP Interday for LLaMA.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
LLaMA	1.521/1.284	1.523/1.285	0.502/0.499	0.003/-0.002	0.504/0.502

As in the ASM MPM setting (see Section 7.7.1), the low number of data points is presumably the main limiting factor, as better results were obtained for the intraday runs, as shown in Table 7.46 and Table 7.47.

Table 7.46: Results for SMP 60min runs.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	**0.524/0.519**	0.020/0.018	0.404/0.408
TransformerXL	0.523/0.518	0.021/0.017	0.413/0.415
LLaMA	0.515/0.515	0.008/0.007	0.572/0.575

Table 7.47: Results for SPP 60min runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.359/0.327	0.359/0.327	**0.518/0.510**	0.040/0.023	0.511/0.507
TransformerXL	0.359/0.327	0.359/0.327	0.515/0.505	0.051/0.033	0.571/0.568
LLaMA	0.359/0.327	0.359/0.327	0.518/0.510	0.038/0.023	0.507/0.503

7.7 ASM Stock2Sentence Results

In the following, all results for the Stock2Sentence ASMs are listed.

7.7.1 Pretraining

The ASM pretraining runs for intraday data have only been performed two times due to hardware constraints. Unless stated otherwise, $|C| = 80$ was used for BERT, T5, and GPT-2, and $|C| = 60$ for TransformerXL and LLaMA. These values were chosen to allow for the largest possible selection of stocks within the constraints imposed by hardware limitations.

MPM The results of the MPM coincide with those of the S2V and the assumptions made there that SMC is a very simple task. MPM is similar to SMC in that it also uses information about future (or past) stock movements to classify the stock under consideration. The expectations of the ASMs are, therefore, higher than the very simple S2V models. As can be seen in Table 7.50, Table 7.49, Table 7.48, the models do not show difficulties in achieving at least the S2V performance. The training shows a similar progression to the S2V models, namely that the model first learns to identify the masked stock (reaches a 50 % plateau in accuracy) and then the correct movement. However, this happens much faster in the ASMs and is usually achieved after the first epoch.

Table 7.49: MPM 1min results. For C all stocks can be samples, i.e. $|\dot{C}| = |C| = 309$ and $\epsilon = 0$

Model	Acc ↑
BERT	**0.996/0.995**
GPT-2	0.995/0.995
T5	0.988/0.988
TransformerXL	0.991/0.991
LLaMA	0.991/0.991

Table 7.50: MPM 60min results. For C all stocks can be samples, i.e. $|C| = 309$ and $\epsilon = 0$

Model	Acc ↑
BERT	0.883/0.878
GPT-2	0.972/0.971
T5	**0.985/0.988**
TransformerXL	0.981/0.984
LLaMA	0.972/0.978

TABLE 7.48: MPM Interday results.

Model	C	Approach	Acc ↑
BERT [6]	S&P-500 🇺🇸	Fixed $\dot{C}$	**0.975/0.975**
GPT-2	S&P-500 🇺🇸	Fixed $\dot{C}$	0.749/0.756
TransformerXL	S&P-500 🇺🇸	Fixed $\dot{C}$	0.750/0.737
LLaMA	S&P-500 🇺🇸	Fixed $\dot{C}$	0.761/0.753
T5 [5]	S&P-500 🇺🇸	Fixed $\dot{C}$	0.962/0.962
BERT	S&P-500 🇺🇸	$\epsilon = 1$	0.699/0.703
GPT-2	S&P-500 🇺🇸	$\epsilon = 1$	0.738/0.792
TransformerXL	S&P-500 🇺🇸	$\epsilon = 1$	0.742/0.751
LLaMA	S&P-500 🇺🇸	$\epsilon = 1$	0.759/0.739
T5	S&P-500 🇺🇸	$\epsilon = 1$	**0.946/0.949**
BERT	S&P-500 🇺🇸	$\zeta = 10$	0.713/0.719
GPT-2	S&P-500 🇺🇸	$\zeta = 10$	0.775/0.778
TransformerXL	S&P-500 🇺🇸	$\zeta = 10$	0.646/0.636
LLaMA	S&P-500 🇺🇸	$\zeta = 10$	0.748/0.754
T5	S&P-500 🇺🇸	$\zeta = 10$	**0.969/0.962**
BERT	S&P-500 🇺🇸	$\zeta = 20$	0.513/0.495
GPT-2	S&P-500 🇺🇸	$\zeta = 20$	**0.783/0.779**
TransformerXL	S&P-500 🇺🇸	$\zeta = 20$	0.751/0.747
LLaMA	S&P-500 🇺🇸	$\zeta = 20$	0.762/0.752
T5	S&P-500 🇺🇸	$\zeta = 20$	0.723/0.724
BERT	All$^{(2010:)}$	Fixed $\dot{C}$	0.823/0.821
GPT-2	All$^{(2010:)}$	Fixed $\dot{C}$	0.716/0.717
TransformerXL	All$^{(2010:)}$	Fixed $\dot{C}$	0.688/0.680
LLaMA	All$^{(2010:)}$	Fixed $\dot{C}$	0.718/0.736
T5	All$^{(2010:)}$	Fixed $\dot{C}$	**0.823/0.824**
BERT	All$^{(2010:)}$	$\epsilon = 1$	**0.863/0.862**
GPT-2	All$^{(2010:)}$	$\epsilon = 1$	0.689/0.717
TransformerXL	All$^{(2010:)}$	$\epsilon = 1$	0.712/0.715
LLaMA	All$^{(2010:)}$	$\epsilon = 1$	0.745/0.749
T5	All$^{(2010:)}$	$\epsilon = 1$	0.854/0.858
BERT	All$^{(2010:)}$	$\epsilon = 5$	**0.786/0.765**
GPT-2	All$^{(2010:)}$	$\epsilon = 5$	0.506/0.521
TransformerXL	All$^{(2010:)}$	$\epsilon = 5$	0.722/0.715
LLaMA	All$^{(2010:)}$	$\epsilon = 5$	0.717/0.722
T5	All$^{(2010:)}$	$\epsilon = 5$	0.707/0.726
BERT	All$^{(2010:)}$	$\zeta = 10$	0.502/0.536
GPT-2	All$^{(2010:)}$	$\zeta = 10$	**0.781/0.800**
TransformerXL	All$^{(2010:)}$	$\zeta = 10$	0.703/0.708
LLaMA	All$^{(2010:)}$	$\zeta = 10$	0.693/0.686
T5	All$^{(2010:)}$	$\zeta = 10$	0.777/0.790
BERT	All$^{(2010:)}$	$\zeta = 20$	0.534/0.506
GPT-2	All$^{(2010:)}$	$\zeta = 20$	**0.830/0.846**
TransformerXL	All$^{(2010:)}$	$\zeta = 20$	0.680/0.677
LLaMA	All$^{(2010:)}$	$\zeta = 20$	0.685/0.685
T5	All$^{(2010:)}$	$\zeta = 20$	0.523/0.544

Additional runs were conducted on All$^{(2010:)}$ for all LLM backbones with $|C| = 898$ and an increase above baseline accuracy was achieved. In order to determine the range in which accuracy gains first emerged, ζ was systematically increased in exponential steps to narrow down the approximate region of interest. Once this interval was identified, a linear progression of values was tested to pinpoint the threshold more precisely. This procedure revealed that improvements above baseline consistently emerged at $\zeta \approx 100$.

By using the ζ approach, all relationships of the stocks cannot be trained at the same time, but at least every stock can be brought into the training set [3].

[3] The condition that each stock occurs in the code was not directly implemented, but sampled randomly. However, it can be stated with $\left[1 - \left(\frac{|C|-|\dot{C}|}{|C|}\right)^{\zeta}\right]^{|C|}$ probability that each stock occurs at least once in the set.

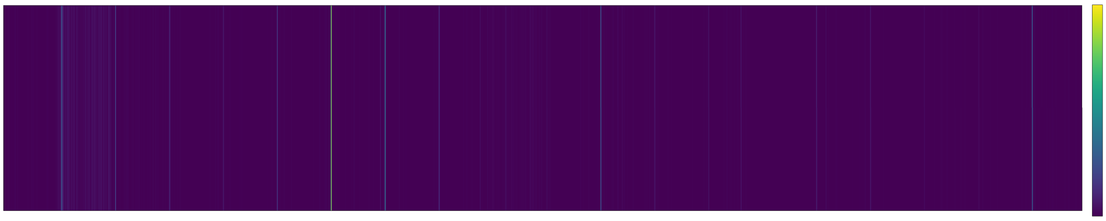


FIGURE 7.17: MPM Relevance visualization of the TMO stock on T5.

The Figure 7.17 was visualized with the Grad-CAM inspired method of [17] and [221] explained in Appendix A.9. Here, however, only one line of the relevance matrix of a randomly chose stock (TMO 🏭) on the corresponding day is visualized, as the relevance was extremely low everywhere else and the display would no longer have worked well. In this layout, the horizontal axis lists the combined company–time pairs, while the vertical axis does not represent another dimension but only serves as the color scale for the relevance values. The visualization shows that only a few localized company–time pairs contribute significantly to the prediction, while most remain near zero relevance. This indicates that the model relies on highly selective signals within the input, reflecting both the sparsity of informative patterns and the noisy nature of financial time series.

Another aspect that was investigated was the impact of pretraining on different time intervals, which appears to provide only limited benefits. Training on higher-frequency intervals generally proves to be sufficient and yields the best performance.

If a model has been pretrained on a 1min dataset, additional training on interday or 60min data does not result in large further performance improvements (especially in downstream performance). Conversely, a model that has been trained on 1min data typically achieves near-perfect performance, making subsequent training on interday or 60min data redundant. It remains uncertain whether this effect is attributable to the larger volume of data in the 1min dataset or the higher temporal resolution. The strong performance observed in the 1min runs with all $c \in C$ supports the former hypothesis. An illustrative example can be found in the BERT run discussed in the following. The BERT run from [6] is utilized. This run achieves an accuracy of 0.997/0.998 in the first epoch, even when further training is conducted on 60min data. Additionally, a triangulated interday run using 1min data was performed, reaching a performance of 0.997/0.998 in the first epoch. However, since this performance is consistently attained by both the 1min and 60min runs and is consistently near-perfect, pretraining across different time intervals was forgone in subsequent experiments. Instead, focus is placed exclusively on the consistently near-perfect intraday runs. Although these results deviate by approximately 5-10% from perfect performance, it was observed in non-tabulated experiments that further pretraining does not yield additional improvements in downstream tasks (SMP/SPP) if the MPM accuracy exceeds 90%. Consequently, further training for these runs was not conducted.

7.7.2 Fine-tuning with SMP as Downstream Task

In the following, the results of the SMP/SPP downstream task will be listed.

Without Additional Pretraining In Table 7.51, Table 7.53, and Table 7.55, the results of the runs conducted on newly initialized models, i.e., without prior pretraining, are presented. The TransformerXL model in Table 7.53 could not create valid results.

In Table 7.52, an approach was explored where 10 stocks were used as prediction targets while 60/80 stocks served as input. For T5, LLaMA, and TransformerXL, results that consistently exceeded baseline performance were not achieved, even when selecting the same set of stocks that performed well in the BERT and GPT-2 runs. It is noteworthy that there was considerable standard deviation, with some runs clearly outperforming the baseline accuracy. The experiment was repeated in Table 7.54 for 60min data.

The interday runs were trained for approximately 500 epochs. Many of these runs exhibit the convergence behavior described in Figure 8.2, typically reaching a local optimum around epoch 150. Notably, the GPT-2 interday model attains its peak performance considerably earlier, around epoch 40. In contrast, the 60min runs generally converge much more rapidly, particularly in the case of runs with $|C| = 10$, which often converge within approximately 10 epochs. Models with larger values of $|C|$ are trained for up to 100 epochs. It is hypothesized that the increased number of stock data per epoch in the intraday runs is responsible for the faster convergence. The 1min runs support this hypothesis, as they typically require only a few epochs to converge. However, this is also influenced by hardware constraints, which render excessively long training times—such as those observed in the interday runs—impractical.

Table 7.51: SMP Interday runs.

Model	$\|C\|$	Acc ↑	MCC ↑	F1 ↑
BERT	80	0.508/0.504	0.017/0.011	0.500/0.499
	10	0.517/0.507	0.030/0.017	0.491/0.504
GPT-2	80	0.503/0.504	0.007/0.007	0.498/0.505
	10	0.520/0.512	0.034/0.017	0.619/0.611
TransformerXL	80	0.501/0.504	0.002/0.009	0.467/0.473
	10	0.501/0.503	0.011/0.015	0.433/0.411
T5	10	**0.526/0.528**	0.044/0.046	0.585/0.585
LLaMA	60	0.503/0.501	0.007/-0.002	0.486/0.485
	10	0.509/0.510	0.011/0.021	0.546/0.551

Table 7.52: Interday SMP runs with 10 stocks as target and 80 as input.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	**0.505/0.510**	0.007/0.012	0.440/0.448
GPT-2	0.506/0.502	0.014/0.005	0.482/0.471

Table 7.53: 60min SMP runs.

Model	$\|C\|$	Acc ↑	MCC ↑	F1 ↑
BERT	80	0.524/0.522	0.009/0.007	0.335/0.332
	80	0.516/0.512	0.002/0.002	0.375/0.378
BERT $+\mathcal{L}_p$	10	0.508/0.500	0.012/0.004	0.539/0.532
GPT-2	80	0.517/0.512	0.005/-0.001	0.366/0.361
	10	0.519/0.526	-0.009/0.013	0.286/0.276
TransformerXL	60	0.524/0.522	0.002/0.002	0.273/0.265
	10	0.530/0.524	0.019/0.011	0.250/0.215
T5	80	0.528/0.526	-0.000/-0.002	0.177/0.165
	10	0.522/0.516	0.012/0.003	0.300/0.281
T5 $+\mathcal{L}_p$	80	0.531/0.538	0.005/0.010	0.623/0.632
T5 $+\mathcal{L}_p$	10	**0.539/0.550**	0.002/0.001	0.650/0.667
LLaMA	60	0.518/0.514	0.007/0.002	0.364/0.369
	10	0.561/0.581	0.001/0.020	0.699/0.719

Table 7.54: 60min SMP runs with 10 stocks as target and 60 as input.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	0.503/0.502	0.007/0.002	0.451/0.447
GPT-2	**0.514/0.517**	0.004/0.004	0.575/0.585
T5	0.512/0.513	0.003/0.003	0.572/0.580
TransformerXL	0.506/0.504	0.004/0.004	0.560/0.571
LLaMA	0.508/0.506	0.004/0.005	0.563/0.574

For the 1min runs, experiments were again conducted with $|C| = 60/80$ and 10 target stocks, but it was found that this significantly reduces performance and decreases stability, resulting in high standard deviation, strong gradient/activation fluctuations, and negative MCC. An example is a BERT run with the result (Acc, MCC, F1-Score) 0.502/0.503 , $-0.003/-0.002$, 0.540/0.545. The presence of a negative MCC strongly suggests that this approach is not viable.

The primary issue with the high-frequency runs is that they frequently result in returns of zero. To address this, experiments were done where the loss was computed exclusively for non-zero returns, i.e., focusing solely on these values during the learning process. For this purpose, a BERT run was conducted (with $|C| = 80$), which produced the following results: 0.552/0.559, 0.011/0.014, 0.684/0.693 (accuracy, MCC, F1). The accuracy for the moving stocks was found to be 0.453/0.448. Consequently, this approach was not pursued further, although it cannot be entirely ruled out that some of the zero returns may have indeed occurred in actual trading. The same experiment was repeated with $|C| = 10$. This yielded similarly strong results, achieving $1 \approx \overline{\underline{\left(\frac{\sum_{i=1}^{|C|} \mathbb{1}(\hat{\mathbf{y}}_i > 0.5)}{\sum_{i=1}^{|C|} \mathbb{1}(\hat{\mathbf{y}}_i \leq 0.5)}\right)}}$ and having accuracy, MCC an F1-score of 0.523/0.529, $0.002/-0.002$, 0.614/0.623. However, the accuracy for the non-zero elements remained relatively low at 0.441/0.419.

Table 7.55: 1min SMP runs.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	**0.564/0.574**	0.030/0.036	0.693/0.706
GPT-2	0.546/0.552	0.006/0.008	0.249/0.247
T5	0.565/0.573	0.004/0.005	0.211/0.221
TransformerXL	0.523/0.526	0.006/0.006	0.387/0.383
LLaMA	0.547/0.553	0.015/0.019	0.290/0.288

ASMs Using S2V Embeddings SMP runs that use S2V embeddings are listed in Table 7.57. The convergence times of the S2V embedding-based runs exhibit a similar pattern to the one described in Section 7.7.2. While these runs generally converge faster—typically within 100 to 200 epochs or fewer for interday data—and in the case of All$^{(2010:)}$ -based S2V embeddings usually between 50 and 100 epochs (with exceptions where training continued beyond the local optimum), they are more frequently affected by class imbalance issues. This pattern continues for the intraday runs: the 60min runs usually converge within approximately 30 epochs, whereas the 1min runs, once again, complete training after only a few epochs.

Table 7.56: SMP Interday on S2V embeddings.

Model	C	$\lvert C \rvert$	Acc ↑	MCC ↑	F1 ↑
BERT	S&P-500 🇺🇸	80	0.500/0.509	0.001/0.021	0.481/0.496
	S&P-500 🇺🇸	10	0.513/0.504	0.026/0.011	0.509/0.508
GPT-2 $+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	80	0.503/0.507	0.004/0.011	0.494/0.495
	S&P-500 🇺🇸	10	0.507/0.497	0.012/-0.005	0.501/0.491
TransformerXL	S&P-500 🇺🇸	80	0.507/0.501	0.014/0.006	0.482/0.478
	S&P-500 🇺🇸	10	0.511/0.512	0.034/0.030	0.591/0.592
T5	S&P-500 🇺🇸	80	0.510/0.510	0.017/0.020	0.534/0.530
	S&P-500 🇺🇸	10	0.509/0.506	0.015/0.015	0.554/0.526
LLaMA	S&P-500 🇺🇸	60	0.502/0.507	0.006/0.010	0.483/0.482
	S&P-500 🇺🇸	10	**0.513/0.518**	0.014/0.026	0.543/0.549
BERT	$\mathrm{All}^{(2010:)}$	80	0.506/0.504	0.012/0.005	0.531/0.526
	$\mathrm{All}^{(2010:)}$	10	0.512/0.509	-0.002/0.023	0.578/0.559
GPT-2	$\mathrm{All}^{(2010:)}$	80	0.509/0.511	0.026/0.024	0.485/0.483
	$\mathrm{All}^{(2010:)}$	10	**0.504/0.522**	0.018/0.049	0.455/0.466
TransformerXL	$\mathrm{All}^{(2010:)}$	80	0.505/0.505	0.011/0.012	0.515/0.510
	$\mathrm{All}^{(2010:)}$	10	0.507/0.512	0.011/0.022	0.511/0.509
T5	$\mathrm{All}^{(2010:)}$	80	0.504/0.504	0.010/0.008	0.495/0.491
	$\mathrm{All}^{(2010:)}$	10	0.522/0.503	0.047/0.007	0.454/0.463
LLaMA	$\mathrm{All}^{(2010:)}$	60	0.531/0.504	0.049/0.018	0.526/0.494
	$\mathrm{All}^{(2010:)}$	10	0.528/0.503	0.048/0.017	0.525/0.492

Table 7.57: SMP 60min on S2V embeddings.

Model	C	$\lvert C \rvert$	Acc ↑	MCC ↑	F1 ↑
BERT	S&P-500 🇺🇸	80	0.519/0.518	0.004/0.009	0.351/0.351
	S&P-500 🇺🇸	10	0.520/0.521	0.002/0.019	0.332/0.327
GPT-2 $+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	80	0.528/0.523	0.015/0.015	0.321/0.326
	S&P-500 🇺🇸	10	0.516/0.517	0.000/0.007	0.346/0.377
TransformerXL	S&P-500 🇺🇸	80	**0.539/0.527**	0.016/0.006	0.246/0.240
TransformerXL $+\mathcal{L}_p+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	10	0.519/0.512	0.018/0.008	0.399/0.393
T5	S&P-500 🇺🇸	80	0.513/0.510	0.005/-0.000	0.394/0.394
	S&P-500 🇺🇸	10	0.519/0.522	0.010/0.015	0.371/0.369
LLaMA $+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	60	0.511/0.509	0.004/-0.000	0.392/0.391
LLaMA $+\mathcal{L}_p+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	10	0.521/0.520	0.011/0.014	0.373/0.367
BERT	$\mathrm{All}^{(2010:)}$	80	0.517/0.514	0.001/-0.001	0.344/0.344
	$\mathrm{All}^{(2010:)}$	10	0.526/0.525	0.020/0.008	0.315/0.294
GPT-2	$\mathrm{All}^{(2010:)}$	80	0.520/0.517	0.006/0.005	0.338/0.341
	$\mathrm{All}^{(2010:)}$	10	0.521/0.524	-0.007/0.015	0.257/0.250
TransformerXL	$\mathrm{All}^{(2010:)}$	80	0.529/0.526	0.005/0.005	0.210/0.195
	$\mathrm{All}^{(2010:)}$	10	**0.525/0.528**	0.003/0.008	0.216/0.195
T5 $+\mathcal{L}_p+F^{\langle \mathrm{R} \rangle}$	$\mathrm{All}^{(2010:)}$	10	0.515/0.510	0.010/0.009	0.416/0.438
	$\mathrm{All}^{(2010:)}$	60	0.519/0.514	0.006/0.001	0.357/0.361
LLaMA	$\mathrm{All}^{(2010:)}$	60	0.518/0.517	0.007/0.009	0.357/0.369
LLaMA $+\mathcal{L}_p+F^{\langle \mathrm{R} \rangle}$	$\mathrm{All}^{(2010:)}$	60	0.509/0.505	0.013/0.008	0.454/0.476

A noteworthy phenomenon was observed when utilizing S2V embeddings, which does not occur when using randomly initialized embeddings. Specifically, for certain models $1 \ll \underline{\overline{\left(\frac{\sum_{i=1}^{|C|} \mathbb{I}(\hat{\mathbf{y}}_i > 0.5)}{\sum_{i=1}^{|C|} \mathbb{I}(\hat{\mathbf{y}}_i \leq 0.5)}\right)}}$ can manifest rapidly. Notably, this effect occurs consistently for the same models and hyperparameter, particularly when employing pretrained S2V embeddings derived from the S&P-500 🇺🇸 index, in contrast to embeddings trained on $\mathrm{All}^{(2010:)}$. Furthermore, it was found that even small values for λ_p can mitigate this process.

ASMs Pretrained on MPM The results for models pretrained on MPM can be found in Table 7.58, Table 7.59, Table 7.60, Table 7.61, Table 7.62 and Table 7.63.

Convergence time is again difficult to assess in this context. Although all interday runs were trained for between 100 and 1000 epochs, many reached a local optimum after only a few dozen epochs. Models pretrained on All$^{(2010:)}$ using the MPM approach typically converge significantly faster—often within just a few epochs, and in some cases, such as with the GPT-2 model, in under 10 epochs. However, this is primarily because these models fail to reach the (assumed) global optimum fast. Otherwise, the ASMs trained on S&P-500 🇺🇸 data would converge just as quickly. Interday runs initialized with pretrained intraday weights tend to be much more stable and usually achieve peak performance after only a few epochs, typically around epoch 10. An exception is the GPT-2 model pretrained on 1min data and fine-tuned for interday and 60min tasks, which was trained for up to 1000 epochs and continued to show performance improvements throughout. A similar phenomenon can be observed for intraday runs initialized with models pretrained on All$^{(2010:)}$ using MPM. The fastest convergence time is again shown by the 1min runs. For GPT-2, a configuration for $|C| > 10$ with stable results could not be found.

Table 7.58: SMP interday runs pretrained on MPM interday runs.

Model	C	$\lvert C\rvert$	Acc ↑	MCC ↑	F1 ↑
BERT	S&P-500 🇺🇸	80	0.511/0.501	0.007/0.007	0.583/0.564
GPT-2	S&P-500 🇺🇸	80	**0.507/0.507**	0.001/0.018	0.561/0.541
TransformerXL	S&P-500 🇺🇸	80	0.522/0.505	0.036/0.002	0.516/0.476
T5	S&P-500 🇺🇸	80	0.507/0.506	0.008/0.010	0.558/0.550
LLaMA	S&P-500 🇺🇸	60	0.508/0.502	0.011/0.003	0.545/0.541
BERT	All$^{(2010:)}$	80	0.509/0.509	0.030/0.015	0.428/0.457
	All$^{(2010:)}$	10	**0.513/0.541**	0.023/0.083	0.507/0.538
GPT-2	All$^{(2010:)}$	10	0.522/0.517	0.059/0.019	0.235/0.208
TransformerXL	All$^{(2010:)}$	10	0.513/0.512	0.024/0.021	0.539/0.525
T5	All$^{(2010:)}$	60	0.506/0.514	0.014/0.017	0.414/0.405
	All$^{(2010:)}$	10	0.525/0.518	0.044/0.041	0.537/0.506
LLaMA	All$^{(2010:)}$	60	0.503/0.503	0.007/0.010	0.499/0.499
	All$^{(2010:)}$	10	0.502/0.509	0.006/0.022	0.501/0.522

TABLE 7.59: SMP interday runs pretrained on MPM intraday runs.

Model	Acc ↑	MCC ↑	F1 ↑
BERT (on 1min weights)	0.501/0.504	-0.002/0.010	0.518/0.503
GPT-2 (on 60min weights)	0.503/0.502	0.007/0.006	0.497/0.495
GPT-2 (on 1min weights)	**0.498/0.526**	-0.009/0.054	0.496/0.529
TransformerXL (on 60min weights)	0.504/0.515	0.010/0.032	0.498/0.506
T5 (on 1min weights)	0.507/0.496	0.014/-0.008	0.498/0.503
LLaMA (on 60min weights)	0.501/0.497	0.003/-0.013	0.464/0.476

TABLE 7.60: SMP 60min runs pretrained on MPM 1min runs.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	0.560/0.571	0.009/0.013	0.693/0.708
GPT-2 $+F^{\langle R \rangle}$	0.520/0.519	0.012/0.017	0.371/0.370
T5	**0.562/0.580**	0.014/0.021	0.696/0.712
TransformerXL	0.515/0.514	0.010/0.014	0.365/0.367
LLaMA	0.512/0.511	0.009/0.013	0.362/0.364

TABLE 7.61: SMP 60min runs pretrained on $\text{All}^{(2010:)}$.

Model	$\|C\|$	Acc ↑	MCC ↑	F1 ↑
BERT	80	0.521/0.519	0.002/0.005	0.348/0.344
	10	0.510/0.512	0.001/0.004	0.343/0.340
GPT-2	80	0.519/0.518	0.002/0.005	0.346/0.342
	10	**0.526/0.522**	-0.003/0.002	0.268/0.262
T5	60	0.521/0.518	0.003/0.001	0.304/0.307
	10	0.515/0.519	0.004/0.013	0.371/0.386
LLaMA $+F^{\langle R \rangle}$	60	0.519/0.514	0.002/0.001	0.302/0.302
	10	0.520/0.517	0.003/0.001	0.303/0.306

TABLE 7.62: SMP 1min runs pretrained on $\text{All}^{(2010:)}$.

Model	$\|C\|$	Acc ↑	MCC ↑	F1 ↑
BERT	80	0.530/0.534	0.001/0.003	0.335/0.333
GPT-2	80	**0.547/0.553**	0.005/0.004	0.223/0.224
TransformerXL $+F^{\langle R \rangle}$	80	0.545/0.551	0.005/0.004	0.221/0.221
T5	60	0.543/0.549	0.004/0.005	0.257/0.257
LLaMA	60	0.546/0.552	0.005/0.004	0.222/0.222

TABLE 7.63: SMP 1min runs pretrained on 1min.

Model	Acc ↑	MCC ↑	F1 ↑
BERT $+F^{\langle R \rangle}$	**0.563/0.573**	0.041/0.049	0.681/0.697
T5	0.520/0.523	0.007/0.007	0.332/0.330

7.7.3 Fine-tuning with SPP as Downstream Task

Without Additional Pretraining The ASM baseline results without pretraining are shown in Table 7.64, Table 7.65 and Table 7.66.

SPP runs converge significantly faster than SMP runs and tend to be much more stable in terms of $\overline{\left(\frac{|\{i|\hat{\mathbf{y}}[i]=0\}|}{|\{i|\hat{\mathbf{y}}[i]=1\}|}\right)}$ or hyperparameter sensitivity. Here, the 100 epoch mark is rarely exceeded even without pretraining.

Table 7.64: SPP Interday runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	1.460/1.274	1.860/0.593	0.507/0.504	0.012/0.007	0.503/0.490
GPT-2	1.464/1.239	1.361/1.547	0.510/0.499	0.021/-0.001	0.505/0.491
T5	1.529/1.303	1.639/0.691	**0.501/0.520**	0.007/0.040	0.502/0.514
TransformerXL	1.498/1.340	1.274/1.027	0.510/0.512	0.022/0.021	0.506/0.504
LLaMA $+F^{\langle\mathrm{R}\rangle}$	1.531/1.274	1.485/0.725	0.499/0.516	0.001/0.031	0.496/0.502

Table 7.65: SPP 60min runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.447/0.390	0.119/0.069	0.531/0.528	0.055/0.048	0.480/0.479
GPT-2 $+F^{\langle\mathrm{R}\rangle}$	0.411/0.349	0.092/0.063	0.528/0.525	0.048/0.044	0.477/0.479
T5	0.358/0.296	0.087/0.039	**0.529/0.530**	0.051/0.053	0.480/0.480
TransformerXL	0.385/0.335	0.141/0.086	0.530/0.529	0.051/0.049	0.477/0.477
LLaMA	0.486/0.439	0.074/0.074	0.528/0.524	0.047/0.040	0.476/0.475

Table 7.66: SPP 1min runs.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.056/0.047	0.003/0.002	**0.572/0.582**	0.122/0.136	0.488/0.488
GPT-2	0.057/0.049	0.003/0.002	0.572/0.581	0.123/0.137	0.492/0.493
T5	0.056/0.047	0.002/0.001	0.566/0.575	0.114/0.128	0.492/0.493
TransformerXL	0.055/0.046	0.002/0.001	0.567/0.576	0.116/0.129	0.493/0.494
LLaMA	0.050/0.041	0.002/0.001	0.572/0.580	0.119/0.131	0.496/0.497

ASMs Using S2V Embeddings SMP runs that use S2V embeddings are listed in Table 7.67 and Table 7.68.

For the S2V embeddings, the SPP-ASMs exhibit an interesting pattern: they converge after more than 100 epochs but also achieve superior performance (in terms of F1-score). In this case, the embeddings likely contributed to avoiding convergence to a local optimum. The intraday runs converge within the expected time frame—typically between 10 and 30 epochs for the 60min setting, and within only a few epochs for the 1min setting.

Table 7.67: SPP interday S2V runs.

Model	C	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	S&P-500 🇺🇸	1.503/1.282	1.287/2.159	0.510/0.509	0.018/0.018	0.516/0.500
GPT-2	S&P-500 🇺🇸	1.500/1.279	1.283/2.154	0.511/0.509	0.019/0.019	0.514/0.498
T5	S&P-500 🇺🇸	1.453/1.257	1.228/1.482	0.511/0.507	0.026/0.014	0.517/0.504
TransformerXL	S&P-500 🇺🇸	1.437/1.259	2.342/0.681	0.500/0.511	0.003/0.021	0.498/0.500
LLaMA	S&P-500 🇺🇸	1.523/1.291	1.814/3.305	**0.507/0.518**	0.014/0.034	0.498/0.498
BERT	All$^{(2010:)}$	1.474/1.269	1.770/2.186	0.496/0.504	-0.006/0.008	0.494/0.497
GPT-2 $+F^{\langle R \rangle}$	All$^{(2010:)}$	1.533/1.326	2.203/0.842	0.500/0.511	0.003/0.019	0.502/0.499
T5	All$^{(2010:)}$	1.512/1.230	1.193/0.589	0.508/0.511	0.019/0.020	0.505/0.489
TransformerXL	All$^{(2010:)}$	1.495/1.192	1.676/0.630	0.511/0.516	0.022/0.035	0.511/0.504
LLaMA	All$^{(2010:)}$	1.507/1.217	1.789/2.208	**0.512/0.521**	0.023/0.037	0.503/0.499

Table 7.68: SPP 60min S2V runs.

Model	C	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	All$^{(2010:)}$	0.398/0.340	0.095/0.065	0.542/0.538	0.072/0.066	0.479/0.480
GPT-2	All$^{(2010:)}$	0.386/0.326	0.171/0.060	0.533/0.530	0.058/0.053	0.477/0.482
T5	All$^{(2010:)}$	0.356/0.303	0.112/0.085	**0.548/0.543**	0.083/0.074	0.482/0.483
TransformerXL	All$^{(2010:)}$	0.354/0.314	0.096/0.052	0.544/0.536	0.076/0.062	0.477/0.478
LLaMA	All$^{(2010:)}$	0.450/0.391	0.107/0.077	0.528/0.525	0.047/0.041	0.475/0.474
BERT	S&P-500 🇺🇸	0.380/0.327	0.121/0.058	0.541/0.533	0.069/0.057	0.476/0.480
GPT-2	S&P-500 🇺🇸	0.380/0.338	0.169/0.061	**0.549/0.537**	0.084/0.061	0.480/0.474
T5	S&P-500 🇺🇸	0.360/0.304	0.085/0.056	0.529/0.526	0.052/0.044	0.481/0.479
TransformerXL	S&P-500 🇺🇸	0.356/0.302	0.081/0.049	0.538/0.533	0.065/0.058	0.477/0.481
LLaMA	S&P-500 🇺🇸	0.449/0.398	0.089/0.051	0.525/0.525	0.043/0.043	0.476/0.479

Table 7.69: SPP 1min S2V runs.

Model	C	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	All$^{(2010:)}$	0.059/0.050	0.003/0.002	0.568/0.576	0.116/0.127	0.489/0.489
GPT-2	All$^{(2010:)}$	0.065/0.056	0.004/0.002	0.566/0.576	0.114/0.129	0.493/0.494
T5	All$^{(2010:)}$	0.068/0.058	0.005/0.002	0.563/0.572	0.111/0.127	0.492/0.492
TransformerXL	All$^{(2010:)}$	0.061/0.052	0.004/0.002	0.566/0.574	0.112/0.128	0.495/0.496
LLaMA	All$^{(2010:)}$	0.068/0.057	0.006/0.003	**0.560/0.589**	0.113/0.129	0.497/0.499
BERT	S&P-500 🇺🇸	0.057/0.048	0.003/0.001	0.564/0.573	0.114/0.127	0.499/0.499
GPT-2	S&P-500 🇺🇸	0.068/0.060	0.003/0.002	0.569/0.578	0.119/0.133	0.495/0.495
T5	S&P-500 🇺🇸	0.062/0.055	0.004/0.002	0.563/0.570	0.114/0.128	0.465/0.498
TransformerXL	S&P-500 🇺🇸	0.061/0.053	0.004/0.002	0.566/0.572	0.113/0.127	0.495/0.497
LLaMA	S&P-500 🇺🇸	0.061/0.052	0.004/0.002	**0.564/0.594**	0.115/0.132	0.499/0.499

ASMs Pretrained on MPM The results for the MPM pretrained SPP runs are in Table 7.74, Table 7.73, Table 7.72, Table 7.71, Table 7.70. The models pretrained on MPM exhibit highly volatile convergence behavior. For MPM runs, a wide range of convergence times is observed, with epoch counts varying from just a few to up to 150. Only the 1min runs consistently show rapid convergence within a few epochs.

Table 7.70: SPP Interday pretrained on Interday.

Model	C	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	S&P-500 🇺🇸	1.541/1.250	1.327/3.309	0.495/0.499	-0.009/-0.000	0.492/0.485
GPT-2 $+F^{\langle \mathrm{R} \rangle}$	S&P-500 🇺🇸	1.507/1.231	0.807/0.483	**0.523/0.523**	0.047/0.047	0.510/0.515
T5	S&P-500 🇺🇸	1.568/1.395	1.342/0.963	0.506/0.504	0.010/0.008	0.501/0.493
TransformerXL	S&P-500 🇺🇸	1.377/1.137	1.129/0.785	0.511/0.492	0.023/-0.013	0.497/0.474
BERT	All$^{(2010:)}$	1.483/1.196	0.980/1.019	0.497/0.505	-0.009/0.008	0.494/0.481
GPT-2	All$^{(2010:)}$	1.447/1.265	1.502/0.634	0.509/0.509	0.021/0.018	0.510/0.502
T5	All$^{(2010:)}$	1.504/1.345	1.952/0.798	0.514/0.503	0.028/0.007	0.510/0.501
TransformerXL	All$^{(2010:)}$	1.579/1.362	1.521/1.319	0.505/0.507	0.011/0.013	0.505/0.500
LLaMA	All$^{(2010:)}$	1.543/1.310	1.111/0.490	**0.485/0.518**	-0.030/0.036	0.459/0.478

In the experimental runs presented in Table 7.70, it is noteworthy how rapidly the maximum SMP accuracy is attained. The accuracy exhibits minimal improvement with continued training, and peak performance is achieved within a few epochs. This observation suggests that the pretraining effectively optimizes the model's convergence time.

Table 7.71: SPP 60min pretrained on 1min.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.431/0.378	0.120/0.061	**0.532/0.526**	0.056/0.045	0.478/0.479
GPT-2	0.394/0.361	0.071/0.061	0.524/0.523	0.044/0.040	0.486/0.483
T5	0.418/0.409	0.049/0.046	0.522/0.521	0.036/0.036	0.469/0.477
TransformerXL	0.347/0.296	0.083/0.061	0.523/0.523	0.038/0.039	0.475/0.477
LLaMA	0.351/0.296	0.093/0.062	0.523/0.523	0.037/0.038	0.473/0.475

Table 7.72: SPP 60min pretrained on All$^{(2010:)}$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.468/0.416	0.080/0.043	**0.528/0.527**	0.047/0.045	0.472/0.476
GPT-2	0.404/0.351	0.118/0.075	0.525/0.522	0.042/0.038	0.475/0.477
T5	0.393/0.348	0.117/0.058	0.527/0.526	0.046/0.045	0.477/0.479
TransformerXL	0.375/0.325	0.091/0.059	0.527/0.523	0.047/0.041	0.478/0.479
LLaMA	0.453/0.384	0.156/0.070	0.527/0.521	0.048/0.033	0.468/0.467

Table 7.73: SPP 1min pretrained on All$^{(2010:)}$.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT $+F^{\langle \mathrm{R} \rangle}$	0.431/0.378	0.120/0.061	0.532/0.526	0.056/0.045	0.478/0.479
T5	0.055/0.047	0.002/0.001	**0.569/0.578**	0.118/0.132	0.491/0.492

Table 7.74: SPP 1min pretrained on 1min.

Model	sMAPE ↓	MAPE ↓	Acc ↑	MCC ↑	F1 ↑
BERT	0.057/0.052	0.005/0.002	0.560/0.570	0.110/0.123	0.493/0.493
GPT-2	0.058/0.050	0.004/0.002	0.564/0.573	0.112/0.125	0.495/0.496
T5	0.053/0.044	0.001/0.001	**0.567/0.577**	0.118/0.135	0.498/0.499
LLaMA	0.056/0.048	0.002/0.001	0.563/0.574	0.116/0.133	0.495/0.495

7.7.4 Further Experiments

The author also tried to integrate E_K into the model by assigning it to $A^{(t)}[j,i] \leftarrow A^{(t)}[j,i] + E_K[\mathbf{k}[i \mod |C|], j]$, without any performance advantage. This is seen positive, as sector information appears implicit in the context-sensitive MPM/S2V embeddings. Given the exploratory nature of the models, future work should investigate eigenvalue-based representations of the time series to mitigate noise. No performance improvement could be achieved here. As already mentioned, experiments with the COMP-head and [COMP]-token to learn the representations were discarded, as the model achieved almost perfect performance in this respect and showed no weaknesses with the S2V representations. This can also be seen very clearly from the fact that the MPM can identify the masked company after just a few training steps, i.e. a plateau of 50% performance is achieved.

Trend Matching Trend matching proved to be a challenging task as in Section 7.5. As this is a new task that has not yet been tested, it is difficult to say whether this is due to the difficulty of the task, the selected hyperparameters or the model architecture. The results/learning are very poor in all implementations, although a learning progress can be seen in the reduction of the loss. For the binary classifications, the accuracy is in the range of 51% to 52% and for the use and prediction of a switched c_i at about $\frac{2}{|C|}$. In this last task, the top-k ($k = 3$) accuracy is also about $\frac{5}{|C|}$ which once again underlines that the model learns at least something. A method was also tested in which the test and validation set were not taken from the time steps following the training set, but from those in the middle of $[1, \mathbb{T}]$, so that the model should theoretically have much easier work, since the complete trend was known except for one time step. Due to the poor results, extensive investigations on TM pretrained models for SMP or SPP were not carried out, as the purpose of pretraining is to perform a simpler task before the actual, complex one. Obviously TM is not a simple task, the model fails on it and it is therefore not useful as pretraining. For the sake of completeness, some test runs were done, and the SMP/SPP performance hardly changed compared to models not pretrained with TM.

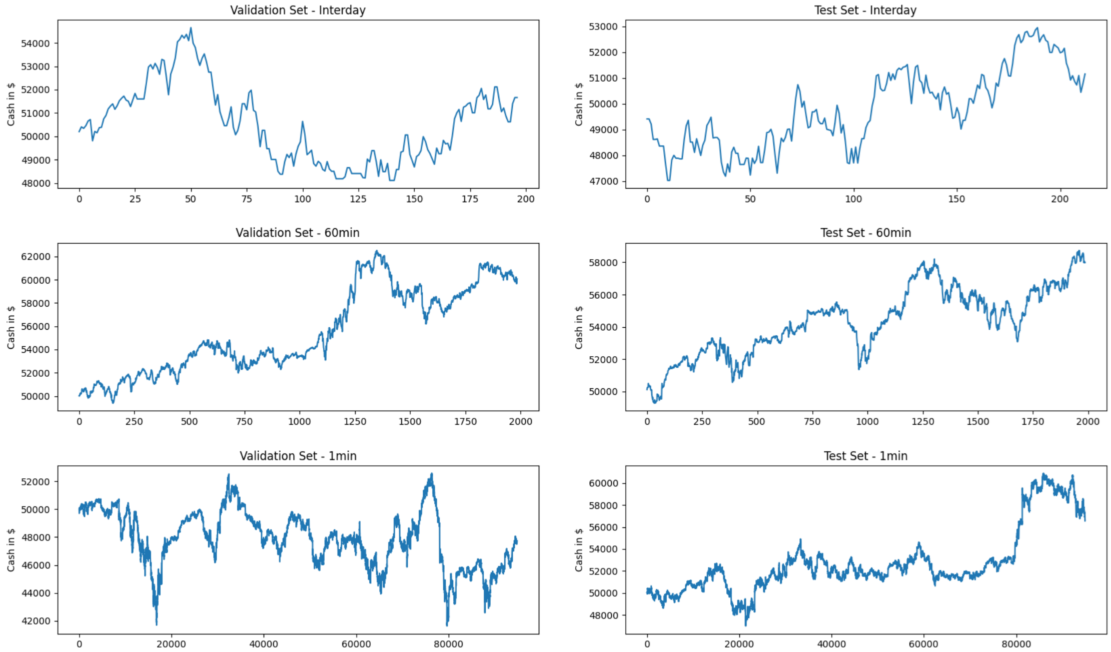


Figure 7.18: ASM Simulation Graphs.

QMSE Integration As mentioned in [223], an attempt was made to integrate the QMSEs $\vec{\mathbf{e}}$. This approach proved largely unsuccessful in the $F^{\langle\text{BM-R}\rangle}$ and $F^{\langle\text{BM-L}\rangle}$ models employed in that work. For the present experiments, in addition to the transformation $\mathbf{e} \leftarrow \mathbf{e}_{\text{QMSE}} \star \mathbf{e}$, the input $A^{(t)} \odot F^{\langle\text{QMSE}\rangle}(f(X^{(t)}))$ was also explored. Both attempts did not bring any visible performance gain and also weight regularization of the self-attention mechanism to focus on the QMSE embeddings more has led to unstable and worse results. Thus, a meaningful integration for $\Pi = e_{\text{QMSE}}$ and a CLM approach adapted for the ASMs with the QMSEs was not found.

7.7.5 Simulation

The results for the simulation can be found in Table 7.75 and in Figure 7.18. The discussion of the implications of the results can be found in Figure 8.2 and Section 8.3.

TABLE 7.75: ASM simulation Results.

Interval	Dataset	CR	Sharp Ratio	IRR	IR	MDD
1min	Val	−4.524	-0.0007	-0.0000	-0.0007	0.2085
1min	Test	13.232	0.0034	0.0000	0.0034	0.1084
60min	Val	21.118	0.0313	0.0001	0.0313	0.1010
60min	Test	15.966	0.0303	0.0001	0.0303	0.0880
Interday	Val	3.320	0.0249	0.0002	0.0249	0.1198
Interday	Test	2.510	0.0172	0.0001	0.0172	0.0603

7.8 ASM Tokenization Based Approach

The results for the ASM tokenization based approach are listed in the following.

7.8.1 MLM

In Table 7.76 are the results for MLM. Due to resource constraints, small Δt and $|C|$ values, i.e., 7 and 5 respectively, were used (as in [222]).

TABLE 7.76: MLM results.

Model	Interval	Acc ↑	Top-5 accuracy ↑
Naive	∼	0.000114	∼
BERT	Interday	0.455/0.392	0.626/0.563
GPT-2	Interday	0.471/0.429	0.621/0.593
T5	Interday	0.723/0.733	**0.819/0.826**
TransformerXL	Interday	0.641/0.647	0.768/0.774
LLaMA	Interday	0.629/0.620	0.762/0.763
BERT	60min	0.305/0.290	0.502/0.480

7.8.2 Fine-Tuning with SMP as a Downstream Task

Experiments with SPP were not conducted due to its poor performance, as already discussed in [222]. It is suspected that the tokenization-based approach is simply not suitable for regression data. The results for the SMP task are presented in the following.

Without Additional Pretraining The SMP results without any additional pretraining can be found in Table 7.78. Intraday runs were not included, as the standard deviation between individual runs was very high, preventing the generation of meaningful or reliable results. Additionally, the resource requirements for such experiments are substantial, making it infeasible to repeat them frequently enough to achieve statistical significance. It is also important to note that, in many cases, baseline performance could not be achieved on one or both evaluation sets, or there were significant discrepancies between the two sets. Overall, as further discussed in Chapter 8, this approach in its current form is not yet sufficiently mature.

Table 7.77: SMP Interday runs.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	0.516/0.501	0.026/-0.005	0.560/0.549
GPT2	0.518/0.496	0.037/-0.008	0.594/0.573
T5	**0.507/0.506**	-0.001/0.034	0.417/0.408
TransformerXL	0.526/0.486	0.031/-0.010	0.604/0.545
LLaMA	0.514/0.493	0.031/-0.010	0.336/0.316

ASMs Pretrained on MPM Results exceeding the baseline were unable to be achieved for the LLaMA model pretrained using MLM. Notably, the pretrained models converge significantly faster, typically within 3 to 10 epochs. In contrast, the non-pretrained models require substantially more training time, with LLaMA needing a minimum of 20 epochs and T5 requiring up to 96 epochs.

Table 7.78: SMP Interday runs pretrained.

Model	Acc ↑	MCC ↑	F1 ↑
BERT	0.494/0.501	-0.011/0.002	0.446/0.492
GPT2	**0.515/0.507**	-0.005/0.015	0.653/0.642
T5	0.490/0.494	-0.015/-0.011	0.528/0.493
TransformerXL	0.520/0.502	0.041/0.003	0.503/0.490

Chapter 8

Discussion

The results are discussed in the following.

8.1 Baseline Results and Implications

The baseline models are first evaluated to establish expectations for downstream tasks, as well as to examine certain challenges and phenomena inherent in stock data (see Section 7.1). The performance of the baseline models highlights the difficulty of SF. For instance, in the case of the $F^{\langle \text{BM-R} \rangle}$ models, no configuration could be identified that achieved any notable learning on the SMP interday data. These phenomena are examples of results that will be observed often in the rest of this chapter. Firstly, a notable difference is observed between the performance on the validation set and the test set. The non-stationary nature of financial markets and the continuously evolving market dynamics result in each dataset—training, validation, and test—exhibiting unique distributions. In such contexts, optimizing hyperparameters solely based on the validation set performance before evaluating the final model on the test set may not accurately reflect the expected performance. Notably, the datasets cover periods such as the COVID-19 crisis, a time characterized by highly unusual market conditions. The limited utility of traditional validation set/test set approach in volatile periods is thereby supported. Given these conditions, it is pragmatic to treat the validation and test sets not merely as sequential temporal datasets but rather as two OOD evaluation sets. This perspective is consistent with evaluation practices for (compositional) generalization across datasets in domains such as vision-and-language, as exemplified by the CLEVR dataset [104], where models are assessed on their ability to generalize beyond the training distribution to OOD evaluation sets. Earlier convergence is often observed on one dataset (e.g., validation set) than on the other (e.g., test set).

The integration of sector, feature, or stock information (E_C and $E_{\mathbb{F}}$) into the models typically failed to yield substantive enhancements, as evidenced by the grid search outcomes in Section 7.1. Aside from Δt, broadly similar results were observed across configurations. This uniformity suggests that the method of incorporation into the baseline models—principally as a learnable scalar bias per stock feature—is suboptimal. In contrast as demonstrated in Section 7.7, the S2V/MPM embeddings already capture the intrinsic characteristics of individual stocks and their intercorrelations to such an extent that the inclusion of sector-level information offered no performance gain in the experiments.

From the intraday datasets (see Table 7.2, Table 7.3), it is indicated that $F^{\langle\text{BM-T}\rangle}$ models benefit from larger data volume or (possibly) lower market efficiency at faster intervals, improving Θ-stability and accuracy.

8.2 Evaluation of the Research Questions

The research questions posed in Section 1.2 are evaluated in the following, beginning with Research Question 1.

> Research Question 1
>
> Which Strategies from the NLP Area can be Adapted for Quantitative Multivariate Stock Price Data and How Can We Use Them?

FIGURE 8.1: Research Question 1 as posed in Section 1.2.

Twofold Approach to the Research Question The first question can be approached from two distinct perspectives: Firstly, there is the broader question of how well adapted strategies can handle quantitative stock or time series data.

Secondly, there is the question of how these strategies can be effectively utilized for SF.

The former question is relatively open-ended and can primarily be addressed through model evaluation, as defined in Section 7.2. This aspect pertains not only to the embeddings themselves but also to the performance observed during the pretraining phase.

The latter question — concerning the practical utilization of these strategies — can be explored through embedding evaluation (as defined in Figure 7.1) using both intrinsic and extrinsic evaluation methods. Furthermore, these considerations can be extended to conceptual frameworks discussed in the subsequent Section 9.2.

S2V and Context-Sensitive ASM Embeddings Following the structure of the classical NLP pipeline, as illustrated in Figure 6.1 and introduced in Chapter 6, the investigation commences with the proposed adapted S2V embeddings (see Section 7.2). Suitability of the adapted W2V models for quantitative stock data is indicated by the good SMC and SPE results.

This conclusion is supported by the outcomes observed during model evaluation (see Section 7.2.1). Furthermore, these results provide a crucial insight: the SMC and SPE tasks appear to be relatively easy, even for the proposed simple S2V models, while remaining similarly unchallenging for more complex architectures such as the ASMs (see Section 7.7.1).

These findings suggest that the core difficulty in stock price forecasting lies in predicting temporal correlations (as defined in [62]). Consequently, this challenge is less concerned with understanding inter-stock correlations or correlations between financial indicators. This observation carries significant implications for the development of future SF models. On a conceptual level, such models should ideally focus on identifying individual stocks for which future price movements can be predicted with a relatively high degree of certainty. Subsequent price movements of other stocks may then be inferred based on these primary predictions.

The markedly superior performance of multivariate models (in S2V) — particularly those incorporating all OHCLV features — compared to univariate, purely time-series-based models substantiates this hypothesis (cf. Table 7.5 and Table 7.4).

Furthermore, this observation underscores the inherent difficulty of accurately predicting individual stock prices in isolation, without accounting for the influence of other stocks. This insight is particularly relevant for the MTM tasks.

The usefulness of the S2V embeddings, on the other hand, is not as straightforward to assess. In most intrinsic evaluation procedures based on absolute scores, their performance appears relatively volatile compared with approaches described in the literature that employ a top-down methodology for embedding construction (see Section 3.0.2). Consequently, the S2V embeddings produce inconsistent results, sometimes achieving strong outcomes while at other times yielding less satisfactory performance. Non-trivial patterns are identifiable in the embeddings via expert/manual analysis (see Section 7.2.4).

Although the identified patterns or clusters are not as distinctly pronounced as those found in the previously mentioned top-down-based architectures (see Section 3.0.2), discernible structures are nonetheless present. Useful patterns are expected to be extractable by domain experts from these embeddings.

The proposed context-sensitive ASM embeddings, which were evaluated in Section 7.2, demonstrate a considerably more stable performance in terms of intrinsic evaluation. Of particular note is the evaluation presented in Table 7.9, which highlights the detection of complex relationships, such as the one identified between SCHW 🪙 and PAYX 🛒. This observation underscores the enhanced capability of ASM embeddings to capture intricate interdependencies within financial data. As observed in the field of NLP, it can be stated that the context-sensitive embeddings derived from ASMs — analogous to the context-sensitive embeddings from LLMs in NLP (e.g. BERT embeddings [40]) — are likely preferable to the S2V embeddings (or W2V embeddings in NLP) (cf. Section 7.7.3 and Section 7.7.2). Superior—and especially more stable—performance in downstream extrinsic evaluation is generally observed for ASM embeddings (cf. Section 7.7.2, Section 7.7.2, Section 7.7.3, Section 7.7.3).

Masking Strategies The successful adaptation of MLM within both the proposed $F^{\langle \mathrm{T} \rangle}$ architectures (Section 7.5.2) and the proposed ASMs (Section 7.7.1) underscores the viability of MLM adaptions for quantitative stock time-series analysis. Focusing initially on $F^{\langle \mathrm{T} \rangle}$-based approaches, the empirical evaluation reveals that MTM exhibits suboptimal performance (see Section 7.5.1). This outcome corroborates findings from S2V experiments and SMC/SME tasks (Section 7.2), wherein classification/estimation of $\mathcal{P}(\mathcal{X} = x_i^{(t)} | \Pi = x_j^{(t)})$, $\forall c_{j \neq i} \in C$ becomes trivial, whereas inference of $X^{(t)}$ absent $\Pi = x_j^{(t)}$, $\forall c_{j \neq i} \in C$ proves challenging. Moreover, inclusion of future time steps $\Pi = \{X^{(t+1)} \ldots X^{(t+\varpi)}\}$ confers negligible classification/estimation benefit for the $F^{\langle \mathrm{T} \rangle}$ models. This is in contrast to the proposed CBOS-X models that benefit from future information (cf. Table 7.5). Since these are significantly less complex, this is another argument against the representations of the stock data in the $F^{\langle \mathrm{T} \rangle}$ models.

Quantitative results demonstrate that spatial dependencies substantially outperform temporal cues in classification/estimation relevance. Although this insight does not improve forecasting, it supports model classes like ASMs that encode spatial structures. $F^{\langle \mathrm{T} \rangle}$-based MTM attains only slight improvements (50.2–51.1% accuracy) over the SMC baseline on intraday datasets, with no bigger advantage for recurrent architectures. MTM's limited performance indicates that pretraining objectives predicated on future-step masking are inappropriate for stock time-series contexts.

MSM similarly underperforms, albeit marginally better than MTM. In contrast, tasks involving the masking of individual price features, as well as S2V-SMC and SPE, demonstrate robust performance. Given that MSM essentially concatenates multiple C-CBOS tasks with supplementary masking—and that even single-feature masking yields poor outcomes—$F^{\langle \mathrm{T} \rangle}$ architectures appear ill-suited for MSM and analogous PPM tasks.

The X-CBOS models (in Section 7.2) employ a very simple architecture, yet they serve as a solid baseline due to their ability to process univariate inputs. Depending on the features used (e.g., "Close" only vs. full OHCLV feature sets), they can already achieve moderately strong accuracies, reaching approximately 0.70 in some configurations.

Pure $F^{\langle \mathrm{T} \rangle}$-based models generally outperform the weaker X-CBOS runs (e.g., those with 0.50–0.55 accuracy), but even in their best configurations, they only marginally surpass the strongest X-CBOS results. Here, as in Section 7.5.1, the intraday runs are disregarded, as they do not allow for a meaningful comparison due to resource constraints and the limited duration of the training processes. In certain settings and models—such as interday data with all OHCLV features and carefully tuned hyperparameters—$F^{\langle \mathrm{T} \rangle}$-models can consistently exceed 0.65–0.70 accuracy (e.g. the $F^{\langle \text{J-M} \rangle}$ in Table 7.24 or most of the $\rho_{\text{heads}} = 1$ models in Section 7.5.1). However, they still fall short of the exceptionally high performance achieved by the ASM models. Due to the strong performance in Section 7.5.1, the multi head attention mechanism or the unsuitability of the time series representation in $F^{\langle \mathrm{L} \rangle}$ is probably responsible for this.

The proposed ASM models in Section 7.7.1, utilizing MPM on OHCLV features, consistently reach accuracies above 0.90 on comparable tasks, occasionally approaching values as high as 0.99 (among the restrictions that were mentioned an Section 8.3). This performance leap clearly demonstrates the superior capability of proposed LLM backbone pretrained models in solving masking tasks efficiently. Overall, these results suggest that $F^{\langle \mathrm{T} \rangle}$-based models are indeed well-suited for masking tasks and often outperform the X-CBOS baseline. However, their advantage is less pronounced than one might expect from high-capacity architectures. Notably, the representation of stock data achieved within the ASM framework remains unmatched by other model types. The hypothesis that the other architectures are not well-suited for implementing pretraining paradigms or serving as the foundation for the quantitative stock foundation models is further substantiated by the observation that the MPM demonstrates near-perfect performance within the ASMs, achieving almost 100% accuracy in numerous instances. This outcome arises despite the fact that the task itself remains identical, and the underlying data are the same, albeit presented in a different format. From this, it can be inferred that while $F^{\langle \mathrm{T} \rangle}$ models are not inherently unsuitable for time series processing, their effectiveness depends on proper alignment with the specific model architecture and data structure (as done in the ASMs).

It is observed that masking is generally suitable for pretraining models in the context of quantitative stock data. However, it becomes evident that the models — like the ASMs — must be specifically designed for this purpose; otherwise, the task cannot be adequately fulfilled.

This finding partially corroborates the necessity of employing context-sensitive embeddings within the proposed ASMs, as well as the internal representation of stock prices and their intercorrelations within these models. As illustrated in Figure 7.17, the mechanism and the relevant time steps and stocks can be seen more clearly. The visualization demonstrates that data points from more distant time steps are of limited relevance. Notably, the corresponding stock itself proves highly influential in the evaluation process, while a particular time step — the second from the left — appears to be especially significant. Furthermore, certain stocks exhibit considerably greater importance than others.

The centrality of distribution shift in MPM pretraining is emphasized (see Chapter 1). The model's near-perfect results on both the validation and test sets illustrate its strong ability to handle OOD data, at least in the pretraining stage.

Trend Matching The TM approach has failed across all models and methodologies, which allows to conclude that NSP cannot be successfully transferred to quantitative stock data — at least not with the current approaches. As outlined in Section 7.7, it is difficult to determine whether this outcome is attributable to the employed models or the task itself, given that this contrastive learning task is novel in the context of stock data. Based on prior work, the difficulty of the task itself is the more plausible cause.

In [223], it was previously observed that the performance of the NNA suggests that similar macroeconomic conditions do not necessarily imply comparable future market movements. In [169], dissimilar sliding windows were blocked during training to prevent the model from being negatively affected by inconsistent patterns. The similarity vector employed for this purpose spanned different time periods, which implicitly indicates that structurally similar periods do not necessarily occur in direct succession.

Considering the persistent distribution shifts in the stock market as well as the previously described observations, it can be inferred that the model, on an abstract level, lacks the capability to identify structural similarities in $\dot{X}$ and $\ddot{X}$ for the purpose of solving the TM task. Instead, it has to compare $\Delta\left(\left\{F : \dot{X} \mapsto \mathbb{1}(\dot{X}^{(\theta_{\mathrm{TM}})} \geq \ddot{X}^{(\omega)})\right\}_{\omega=1}^{\theta_{\mathrm{TM}}} || \left\{\ddot{X}^{(t)}\right\}_{t=1}^{\theta_{\mathrm{TM}}}\right)$ (following the notation in Section 6.9.2). This essentially corresponds to a generative task, which has already been identified as particularly challenging in Chapter 2.

Given the difficulties the models exhibit even when predicting $X^{(t+1)}$, it is reasonable to assume that the quasi-predictive classification of longer sequences within the context of contrastive learning poses an even greater challenge for the model. As compensation for the absence of the macroeconomic states intended by the TM approach, as outlined in Chapter 1, QMSEs were developed and integrated into the models. Regarding the NLP perspective, NSP was unable to be successfully adapted as a pretraining task.

All$^{(2010:)}$ Pretraining Dataset For the ASM, QMSEs, and S2V models, it was possible to utilize the All$^{(2010:)}$ dataset as a pretraining dataset, analogous to the large unlabeled text corpora typically employed for pretraining LLM models. This is useful for two main reasons. First, due to its size compared to the S&P-500 🇺🇸 dataset and the absence of the first ten years of data, it is more up-to-date. However, it therefore lacks insights into earlier market dynamics. For some S2V models, the All dataset was additionally utilized, meaning data starting before 2010 was included, resulting in a significantly smaller $|C|$. Second, the All$^{(2010:)}$ dataset is noteworthy for its pronounced (national) market heterogeneity.

In the evaluation of the S2V C-CBOS models, no significant performance differences between the datasets are observed, as shown in Table 7.4. In the X-CBOS models, the model appears to benefit from a differentiated and broader data heterogeneity for individual univariate stocks, as evidenced by the performance of the All$^{(2010:)}$ run in Table 7.5. Of course, the model only benefits from heterogeneity in the embedding representations and not at a spatial level. Regarding the evaluation by country (Table 7.10 and Table 7.11), the S2V embeddings, as observed in all embedding evaluations, do not achieve strong absolute performance scores. However, as illustrated in Figure 7.6, S2V structures relevant to international markets were able to be identified.

For the overall evaluation of the QMSEs presented in Section 7.4, significantly poorer performance is observed for the models trained on the All$^{(2010:)}$ dataset, with few exceptions. This may possibly be attributed to the higher complexity inherent in this dataset, as here significantly more stocks need to be represented in the low dimensional QMSE representations.

Regarding the ASM pretraining task, model performance on the All$^{(2010:)}$ dataset improves particularly when $|C|$ is limited (i.e., ϵ or ζ approaches). However, the performance differences for larger $|C|$ values, such as $\zeta = 20$, reveal the dataset size limitations inherent to the All$^{(2010:)}$ dataset. These limitations have already been discussed in Section 8.3.

In terms of pretrained S2V embeddings (cf. Section 7.7.3 and Section 7.7.2), no single pretraining dataset (S&P-500 🇺🇸 vs. All$^{(2010:)}$) consistently outperforms the other across all models and metrics. Notably, BERT shows a slight tendency to benefit from pretraining on the All$^{(2010:)}$ dataset, whereas T5, in contrast, appears to gain marginal improvements from pretraining on S&P-500 🇺🇸. For the decoder-only models—GPT-2, TransformerXL, and LLaMA—the results vary by metric and training setup, suggesting that while the choice of pretraining corpus has an influence, it does not solely determine model performance. It is also worth mentioning that the S2V embeddings, which were trained on interday data, exhibited extreme fluctuations when applied to intraday data in SMP (see Table 7.57). This, in particular, led to an escalation of the phenomena described in Section 8.3. Mitigating these issues was only achievable through the use of λ_p and/or $F^{\langle \mathrm{R} \rangle}$. For ASMs on the SMP task Section 7.7.2 and Section 7.7.2 can be compared. TransformerXL shows the most consistent F1-score gains when using S2V, especially in interday settings, while T5 can also achieve strong F1-score improvements, though results depend heavily on hyperparameters. GPT-2 and LLaMA tend to improve accuracy/MCC but often at the cost of F1-score, with GPT-2 showing frequent F1-score drops. BERT shows moderate and mixed changes across all metrics. The

All$^{(2010:)}$ embedding set produces more pronounced effects (positive and negative) than S&P-500 🇺🇸, making S2V embeddings most useful when optimizing for F1, particularly with TransformerXL or T5 under the right conditions.

In the SPP (cf. Section 7.7.3 and Section 7.7.3) experiments, All$^{(2010:)}$ and S&P-500 🇺🇸 show no clear overall winner, but their effects differ by model. T5 reacts more variably—sometimes favoring All$^{(2010:)}$ for error metrics, but often performing better with S&P-500 🇺🇸 for classification. GPT-2 and T5 generally show more stable MCC/accuracy gains with S&P-500 🇺🇸, while LLaMA gains modestly from All$^{(2010:)}$, especially at 60 minutes, though it fluctuates more interday with S&P-500 🇺🇸. Overall, All$^{(2010:)}$ induces stronger but more volatile effects, whereas S&P-500 🇺🇸 leads to steadier classification improvements, especially for GPT-2 and T5. The comparative analysis indicates that the All$^{(2010:)}$ embeddings tend to induce larger variations in F1-score performance while the S&P-500 🇺🇸 embeddings yield more consistent gains in accuracy and MCC. Consequently, the choice between these embedding sets should be guided by the specific performance metrics and model architectures of interest, rather than a one-size-fits-all approach.

A key question concerns how much ASMs benefit from pretraining on heterogeneous markets. As discussed in Figure 8.2, the All$^{(2010:)}$ dataset leads to improved performance in MPM pretraining only in a minority of cases, and even then, the gains are marginal. This observation holds particularly true for intraday data (see Table 7.61, Table 7.62, Table 7.72 and Table 7.74). This can presumably be attributed to the limited amount of data in All$^{(2010:)}$, which likely prevents the model from learning stock relationships as comprehensively as in the S&P-500 🇺🇸-based MPM pretraining.

First, the SMP interday runs are examined in Table 7.58. It is initially noticeable that some models only produced usable results with $|C| = 10$ when pretraining was performed on All$^{(2010:)}$ (GPT-2, TransformerXL). In all cases where larger values of $|C| = 60/80$ were used, performance dropped (sometimes significantly) on All$^{(2010:)}$ compared to S&P-500 🇺🇸. However, the peak performance for all

models pretrained on All$^{(2010:)}$ was achieved in runs with $|C| = 10$. For runs pretrained on S&P-500 🇺🇸, no usable performance could be achieved with $|C| = 10$.
Next, the intraday runs are compared, starting with the 60min runs (cf. Table 7.61 and Table 7.60). Except for LLaMA, a similar pattern is observed: performance decreases slightly to significantly on larger $|C|$ values when pretraining was performed on All$^{(2010:)}$. However, performance on the $|C| = 10$ runs does not surpass the performance on larger $|C|$ values or on the S&P-500 🇺🇸 runs, with the exception of GPT-2. It is also striking how poor the F1-score becomes on the runs pretrained on All$^{(2010:)}$, especially when compared to the top-performing BERT and T5 models. For the 1min runs, many of the S&P-500 🇺🇸 models yielded no usable results (see Table 7.63), and even T5 could not reach the performance of the models pretrained on All$^{(2010:)}$ (see Table 7.62). However, the BERT model pretrained on S&P-500 🇺🇸 could not be outperformed by any of the All$^{(2010:)}$ runs in Table 7.62.

Overall, pretraining on the All$^{(2010:)}$ dataset proved beneficial for SMP only in a few cases and must therefore be considered rather risky. In the interday SPP in Table 7.70, the results are mixed and it depends on the model which dataset is more suitable. For the SPP intraday runs, the 60min runs are considered first, comparing Table 7.71 with Table 7.72. Here, if performance differences exist, they are marginal—significantly smaller than in the interday runs—and again model-dependent. For the 1min runs in Table 7.73 and Table 7.74, a pattern similar to the SMP runs is observed. Most models pretrained on S&P-500 🇺🇸 perform rather poorly, with the exception of the T5 model, which outperforms all others. The relatively well-performing BERT model pretrained on S&P-500 🇺🇸 could not match the performance of the All$^{(2010:)}$ runs.

In conclusion, unfortunately, no strong advantage of the heterogeneous All$^{(2010:)}$ dataset can be identified—at least under the given limitations. However, the use of the All$^{(2010:)}$ dataset significantly reduces the risk that models do not produce usable results for intraday runs.

Notably, the models pretrained on the All$^{(2010:)}$ dataset exhibit significantly faster convergence times in most cases, ranging between 9 and 70 epochs. In contrast, the models trained on the S&P-500 🇺🇸 dataset required several hundred epochs. Also the phenomenon described in Figure 8.2 more frequently occurred.

Doc2Vec Adaption The evaluation of the proposed Doc2Vec adaptation strategy has been discussed in detail in [223] and in Section 7.4. This adaptation can be regarded as a partially successful implementation of a NLP strategy. Its success is particularly evident in the positive model evaluation as well as its effective deployment in various other model architectures. Among other contributing factors, this can be attributed to the function of QMSEs, which—as discussed in the preceding section—provide a structural foundation for integrating macroeconomic information. In this context, the integration of QMSEs as a learning regularizer also plays a role, albeit with comparatively limited success.

The proposed adaptation from the NLP domain must be viewed self-critically as relatively broad in scope, given that the use of established NLP models such as Sentence-BERT [194] or Skip-Thought [109] proved unsuccessful, and the application of AEs to generate dense vector representations of text passages is rather uncommon in NLP.

The relevance of QMSEs, particularly their utility as indicators of unusual market dynamics with parameter d, as discussed in Section 6.7, becomes more pronounced in the context of downstream evaluation and simulation, as elaborated in Figure 8.2.

The integration of QMSEs as a learning regularizer has, in some cases, improved model training, as described in Section 7.7 and Section 7.5. However, as previously noted in [223], this effect is more pronounced in terms of influencing the standard deviation of training performance rather than enhancing overall model accuracy. This effect appears particularly effective in 'unstable' models such as the S2V-based ASMs in high-frequency time intervals, as demonstrated in Table 7.57.

Tokenization and Adapted LLM Models The proposed utilization of LLMs has been analyzed from three distinct perspectives, as discussed in [222]. Among the proposed approaches, embedding-based methods demonstrate characteristics closely resembling those of the $F^{\langle\text{BM-T}\rangle}$ model, differing primarily in the specific architectural and functional properties inherent to each respective LLM.

As shown in Section 7.6, it becomes evident that models designed for training with extensive datasets and large text corpora exhibit significant limitations when applied to interday stock data. These limitations suggest that the inherent structure and training paradigms of such models may not be well-suited for capturing the temporal dependencies and volatility patterns characteristic of financial time series data at this granularity.

Moreover, the decoder-only GPT-2 model has proven entirely unsuitable for this particular application. This outcome may be attributed to GPT-2's autoregressive decoding strategy, which fails to effectively model the complex temporal relationships required for accurate interday predictive tasks. Conversely, both the LLaMA and TransformerXL models have demonstrated promising performance in this context. Notably, LLaMA emerges as the only proposed adapted model that consistently yields reliable results for interday stock data. This suggests that LLaMA's architecture and training objectives align more effectively with the characteristics of financial time series data.

Using Stock2Sentence-ASMs, the most promising approach to adapting LLMs as foundation models for time-series processing, specifically for quantitative stock time series, is proposed.

The tokenization method proposed in Section 7.8 proves effective in several respects. The MLM pretraining results, presented in Table 7.76, are noteworthy: approximately 80% of input tokens represent numeric values (with the remaining 20% corresponding to ticker or feature symbols). Consequently, the model accurately reproduces a substantial proportion of regression targets, which is all the more impressive given that each digit has to be guest by the model.

However, the SMP token-based sequence modeling approach entails inherent risks due to its substantial computational demands, limiting feasible values for the number of $|C|$ and Δt. As previously emphasized, incorporating a broad cross-section of stocks is essential for capturing inter-stock correlations; a small C risks omitting stocks that are crucial for predictive performance. This constraint explains the high variance across runs and indicates that performance differences may result from stock set sample selection.

Overall, pretraining exerts minimal positive impact on forecasting accuracy. The findings in Section 7.8.2 align with the authors previous one in [222].

The tokenization-based models achieve high F1-scores and show numerical stability, suggesting robustness to non-stationarity.

Recurrent / Long Architectures as Cutting Edge NLP Research Direction As discussed in Chapter 1, the processing of long sequences remains a major challenge in NLP, particularly in the context of transformer-based architectures. To address this issue in the domain of financial time series, the use of recurrent transformer architectures as a potential solution when applied to stock market data was proposed (Section 6.8). This approach allows to formulate hypotheses about the extent to which SF models may benefit from recent advances in NLP research.

In [224], improved SPP performance associated with an increased context window size Δt for high-frequency (1min interval) stock data was observed. However, this improvement could not be replicated in the experiments in Section 7.5.2, with the new padding methods or for longer interval datasets, such as 60min or interday data (see Table 7.32), which are also characterized by significantly smaller data volumes. The limited availability of data, as further discussed in Section 8.3, appears to play a critical role in this context. Nevertheless, neither SMP nor SPP models consistently demonstrate improved performance across all temporal intervals with larger values of Δt (see Tables 7.31 and 7.32). A notable advantage of some proposed recurrent models, such as the $F^{\langle \text{L-M} \rangle}$ in Table 7.18, is the time it takes to converge to untrained data after a few epochs. In particular, Table 7.19 also shows the difficulty in pretraining some models to process the recurrence, if one compares the performance with $F^{\langle \text{T} \rangle}$.

Hierarchical Processing For SMP, however, the results are more nuanced (see Table 7.15 and Table 7.3). No consistent superiority of the CWRNN models over the baselines can be identified across all time intervals and metrics. On interday and 60min data, performance differences between both model classes are minimal, with neither clearly outperforming the other. While the CWRNN models occasionally show a higher F1-score, the baselines tend to yield slightly higher values for accuracy and MCC, particularly on the 1min data. This suggests that the benefits of hierarchical processing might be less pronounced for the SMP task or are overshadowed by other modeling aspects such as temporal granularity or class imbalance.

Taken together, these findings support the hypothesis that hierarchical temporal modeling contributes to performance improvements, especially for SPP tasks at coarser time resolutions. This makes sense because many of the periodic patterns mentioned in Chapter 1 occur mainly at coarse granular frequencies.

Research Question 2

To What Extent can Adapted Strategies Contribute to Improving Prediction?

Figure 8.2: Research Question 2 as posed in Section 1.2.

To address this question, it is necessary to consider to what extent performance can be improved through adapted strategies, such as pretraining, the impact of pretraining on convergence time, the adaptation of CLM in various forms, and overall performance enhancement.

Performance Gain by Pretraining The first question to address is to what extent the absolute performance of ASM and $F^{\langle \mathrm{T} \rangle}$ models can be improved through pretraining.

SMP is considered first for the $F^{\langle \mathrm{T} \rangle}$ models. The proposed $F^{\langle \text{J-M} \rangle}$ model is identified as the strongest in SMP without pretraining in interday as seen in Table 7.33. Performance surpasses baseline accuracy (clearly) on the test set only through pretraining, as shown in Table 7.39. This is considered positive, as pretraining enables the model to learn the actual SMP task instead of overfitting to the validation set. However, the F1-score decreases, a phenomenon observed frequently, suggesting that pretraining tends to support the use of winner and loser stock strategy (see

Section 8.3) for some models. The opposite can often be observed in the ASMs discussed below. The proposed $F^{\langle\text{E-M}\rangle}$ model is slightly negatively affected by pretraining, although it still belongs to the stronger SMP models even without it. The $F^{\langle\text{E-M}\rangle}$ model is also the only one that delivers relevant and table-worthy results on the 60min runs with pretrained data. Although the accuracy remains comparable in Table 7.40 to the non-pretrained version in Table 7.34, the F1-score improves significantly. This stands in contrast to the interday results and is particularly noteworthy due to the dataset imbalance described in Chapter 4. The same phenomenon is also observed for the $F^{\langle\text{J-M}\rangle}$ model in 1min runs, where accuracy is comparable (slightly lower), but the F1-score again increases (cf. Table 7.35 and Table 7.41).

The SPP task is now addressed for the $F^{\langle\text{T}\rangle}$ models. As shown in Table 7.36, extremely strong performance is achieved for interday runs on the validation set, which could not be matched in any other model configuration. On the test set, the performance remains acceptable. As before, test set performance (which is presumably less similar to the training set distribution) is improved through pretraining, as presented in Table 7.42, especially for the $F^{\langle\text{J-M}\rangle}$ model, while it slightly decreases for other models. For the intraday 60min data, a similar pattern to SMP is observed, and the performance on pretrained data in Table 7.43 yields a slightly better F1-score compared to Table 7.37. For the 1min data, little to no improvement is observed, with slight degradations due to pretraining (cf. Table 7.38 and Table 7.44).

For SMP, pretraining shows a slightly positive effect with the appropriate models, particularly in handling imbalanced classes in intraday data, or no effect overall. For SPP, a similar pattern is observed, though more pronounced for interday data and less so for intraday data (or not at all for 1min data).

When compared to ASMs, the proposed $F^{\langle\text{T}\rangle}$ models generally lag slightly in terms of classification performance. In certain cases, such as the 60min SMP task, transformer models like $F^{\langle\text{E-M}\rangle}$ or $F^{\langle\text{J-M}\rangle}$, when equipped with MPM/MFM pretraining, can approach or even match the accuracy levels achieved by ASMs such as T5 or

BERT. Although the performance improvements observed in transformer models remain slightly below those of the top-performing ASMs, the results clearly underscore the consistent benefits of pretraining across both model families. A persistent limitation of transformer architectures, however, lies in their lack of structural extensibility.

In SMP interday without pretraining, ASMs generally show better performance, especially on the test set, which implies that better generalization is achieved (cf. Table 7.51 and Table 7.33). The best ASM runs, such as BERT (with $|C| = 10$), GPT-2 (with $|C| = 10$), and T5, cannot be matched by $F^{\langle \mathrm{T} \rangle}$.

The effects of pretraining on ASMs (cf. Table 7.51 with Table 7.58 and Table 7.59) are found to be positive for interday runs on the S&P-500 🇺🇸 dataset, used as the pretraining dataset in all cases except for the (already extremely strong) T5 model. A particular impact is observed on the F1-score, which helps to counteract class imbalance and allows the actual SMP problem to be solved without employing the winner-loser stock strategy. When pretraining is conducted on the All$^{(2010:)}$ dataset, performance gains are found to be mixed. Except for the TransformerXL model or BERT (with $|C| = 80$), improvements are observed either on the test set or the validation set. The T5 model still remains unmatched in terms of performance. The F1-score generally improves in every case, although not as strongly as with the other pretraining. The performance on pretrained intraday runs is considerably weaker than on interday runs and has been discussed in Section 8.3. Significant improvements are observed only for TransformerXL.

For the SMP intraday runs, attention is first directed toward the 60min runs that were pretrained on the 1min intraday runs (cf. Table 7.60 and Table 7.53). For BERT, the F1-score is found to be very poor except for the $|C| = 10$ runs. As with interday runs, pretraining on the same dataset appears to help guide the training in a more favorable direction, which proves to be highly effective for BERT. Overall accuracy is also improved when compared to the $+\mathcal{L}_p$-based model. Dataset bias is better resisted, and the learned weights seem to represent a (local) optimum, preventing SGD from converging in a direction overly influenced by dataset bias. For the decoder-only models GPT-2 and TransformerXL, a notable increase in F1-score by approximately 0.1 is observed in both cases, although absolute performance remains poor. In both cases, however, accuracy is negatively affected. The encoder-decoder T5 model again achieves the best performance, with $+\mathcal{L}_p$ yielding even better F1-scores, and pretraining resulting in a level of performance not reached by any other model, aside from the LLaMA model with $|C| = 10$. For the All$^{(2010:)}$ dataset, performance mostly decreases, and the F1-score in particular suffers (see Table 7.61). In cases where performance increases are observed, such gains rarely exceed 0.02 accuracy points. It is presumed that, due to the volume limitation of All$^{(2010:)}$, the SGD process fails to reach a region where the stabilizing effect of pretraining becomes active.

For 1min intraday runs, a similar pattern is observed with All$^{(2010:)}$, and both accuracy and F1-score degrade (cf. Table 7.62 and Table 7.55). The rare performance gains are deemed negligible. For the other pretrained 1min runs in Table 7.63, a similar trend is noted, and the outcome cannot be attributed to the All$^{(2010:)}$ dataset. It is presumed that the volume of data at 1min intervals is so large that any influence of pretraining becomes irrelevant. For the models not listed in Table 7.63, no stable results could be produced.

Attention is now turned to SPP: SPP is generally found to be significantly more stable, and recourse to λ_p is seldom required. The F1-score is observed to be better than in SMP, which is particularly noteworthy given the high class imbalance in intraday runs. This is likely because the MSE loss is implicitly more suitable for SMP than BCE. The sMAPE is found to be worse in almost all cases after pretraining, which can be attributed to the fact that, as explained in Chapter 4, the SMP metrics accuracy, F1, and MCC are optimized, as these are ultimately decisive; regression metrics are therefore mentioned only in passing. In the interday runs that were pretrained on the S&P-500 🇺🇸 dataset (see Table 7.70), SMP performance deteriorates in nearly all cases, and no usable result could be obtained for LLaMA. GPT-2 is the only exception, showing a significant benefit from pretraining and achieving unmatched SMP performance on interday data. Pretraining on the All$^{(2010:)}$ dataset yields a similar result (with GPT-2 again being the only model to benefit substantially); where performance gains occur, they are marginal.

For intraday runs at 60min (see table 7.71 and Table 7.72), performance is mostly slightly worse, and any gains are negligible (compared with those in Table 7.66). On 1min data without pretraining (see Table 7.66), post-pretraining performance is poor for most models, especially after pretraining on S&P-500 🇺🇸 data (see Table 7.73), with BERT performing particularly poorly. After pretraining on the All$^{(2010:)}$ dataset, only T5 achieves slight performance improvements (see Table 7.74).

In summary, pretraining is not found to be beneficial for SPP in the ASMs. Performance gains are usually minimal, and in most cases, performance is degraded. This may be due to the fact that pretraining is based on an SMP task, and the dynamics learned during this process only confuse the model in the SPP context. It is also possible that SMP performance is already so strong that no additional improvement is achievable through pretraining.

The choice of the pretraining dataset is crucial in this regard. A broad dataset such as All$^{(2010:)}$ particularly enhances T5's generalization capabilities, whereas the S&P-500 🇺🇸 dataset enables more targeted improvements in the SPP task for models like GPT-2 and TransformerXL. Pretraining effectively reduces dataset bias and stabilizes class predictions, which is reflected in significantly improved F1-scores.

Notably, BERT and TransformerXL exhibit more stable classification and regression performance in interday and 60min environments when pretrained on appropriate financial data. Furthermore, pretraining accelerates convergence, as market-related patterns are already internalized beforehand, significantly reducing training time. Despite these advantages, performance improvements remain incremental, reflecting the ongoing volatility and noise in market data.

SMP tends to benefit more from pretraining (especially with interday data or slight class imbalance) than SPP. For intraday data (60min), models can benefit particularly in terms of the F1-score when pretrained. The accuracy often remains unchanged or drops slightly. At 1min, few advantages can be seen. ASMs typically show higher baseline performance than $F^{\langle \mathrm{T} \rangle}$-models and benefit more clearly from pretraining. $F^{\langle \mathrm{T} \rangle}$-models can benefit from pretraining in SMP (e.g. $F^{\langle \text{E-M} \rangle}, F^{\langle \text{J-M} \rangle}$), though they are generally somewhat more inconsistent than ASMs. For SPP, the benefit of pretraining in $F^{\langle \mathrm{T} \rangle}$-models is generally limited.

Using S&P-500 🇺🇸 as a pretraining dataset often works better than All$^{(2010:)}$, due to larger data volume and higher representativeness. Overall, it appears that pretraining is more worthwhile for SMP (for both $F^{\langle \mathrm{T} \rangle}$-models and ASMs) than for SPP. For SPP, the gains are generally smaller—some models even lose performance, possibly because pretraining does not align well with the target task or because SPP performance is already strong and leaves less room for improvement. Despite these findings, the ASMs remain the preferred choice for further analysis, given their stronger baseline performance, clear benefits from pretraining, and greater extensibility. They also exhibit characteristics more closely aligned with foundation models such as LLMs.

Convergence Time through Pretraining One of the key considerations—alongside the general linguistic capabilities and universality of LLMs—is their convergence time. Pre-trained LLMs enable resource-efficient fine-tuning, often requiring only a handful of epochs to achieve satisfactory performance. This paradigm has contributed to the great popularity and success of LLMs in the ML community as the resources required for the use of speech models in specific contexts have been drastically reduced. This is an important step towards making NLP more accessible [254].

Whether the sole motivation is represented by this, or whether an impact on performance is generally exerted by foundation models, is rarely debated, and a mixed picture is presented by the few existing studies; cf. [89] and [254] in NLP, or [83] in CV. Nevertheless, the paradigm has become so deeply rooted in SOTA practices that the debate can be considered irrelevant. However, based on our previous results, it is found that performance in the SF setting is not necessarily benefited by pretraining. This raises the question of whether at least convergence time and resource requirements can be reduced.

Applying the proposed paradigm to SF presents unique challenges. In SF tasks, expected model accuracy tends to remain low, which in turn constrains the optimization window before overfitting occurs. Consequently, practitioners must employ very small learning rates and extended training durations to avoid overfitting, thereby negating some of the convergence-time advantages.

By leveraging pretraining in ASMs, the time required to reach a local optimum can be substantially reduced. However, in SF models this rapid convergence is not necessarily desirable. Because the optimization window is extremely narrow—owing to the low expected accuracy pre-trained models quickly reach a suboptimal local minimum or overshoot it altogether. In practice, it can therefore be more effective to train the model from scratch, since fine-tuning a pre-trained model often prevents it from finding the true optimum within this constrained Θ space. For the approaches and datasets presented in this thesis, the training times are also limited and are not comparable to those of classical LLMs, which is why the convergence time is not a worthwhile argument under the paradigms presented here.

Another illustrative example is shown in Figure 8.3. On MPM pre-trained ASMs typically reach a reasonably good local optimum very quickly, as noted previously. However, continued training beyond this point—often for many additional epochs—yields only marginal improvements toward what appears to be a global optimum. Critically, the loss curve provides no clear signal that further training is beneficial, and the practitioner cannot easily determine when to stop. Consequently, without careful monitoring, training may either stop too early or continue longer than necessary with minimal benefit.

In summary, although pretraining typically reduces convergence time, this advantage is not necessarily desirable in SF applications. The expected overall performance, optimization landscape, and numerical characteristics of SF differ substantially from those of NLP tasks. Consequently, focusing too much on rapid convergence can reduce model effectiveness, as the constrained optimization landscape and distinctive loss dynamics in financial forecasting make convergence time less important than achieving robust generalization and general performance. The phenomena shown here also apply to S2V embedding based runs, which tend to produce ‘destroyed’ Θ even faster.

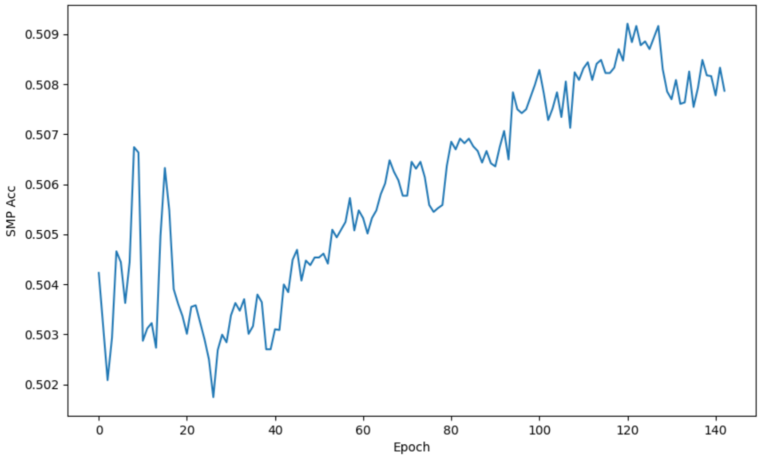


FIGURE 8.3: Example of longer finetuning for better performance.

CLM Adaption The adaptation of CLM must unfortunately be regarded as largely unsuccessful. With the exception of the integration of textual data—which was only briefly explored in the experiments and is discussed in more detail in Section 9.2—the inclusion of QMSEs as Π yielded no positive effect on model performance, consistent with the findings in [223]. Moreover, masking within the transformer architecture, which in some cases employs $\Pi = X^{(t'>t)}$ or $\Pi = \mathbb{M}^{(t)}$ (as defined in Section 7.2), also failed to improve results and can likewise be viewed as an adaptation of CLM. Overall, time series data does not appear to be suitable as a modality for the CLM adaptations. It works better if the time series has a formatting that is more optimized for the spatio-temporal aspect, as in the Stock2Vec/ASM MPM.

General Performance The simulation of the best-performing models, presented in Section 7.7 and illustrated in Figure 7.18, indicates that ASMs can produce profitable trading strategies. Notably, these models demonstrate strong suitability across different time intervals: they perform exceptionally well on interday data—where markets are generally more efficient—yet exhibit comparatively weak performance on 1min data. A particularly important observation, first noted in Section 8.1, is the pronounced discrepancy between results on the validation set and those on the test set. This divergence underscores the challenges of model

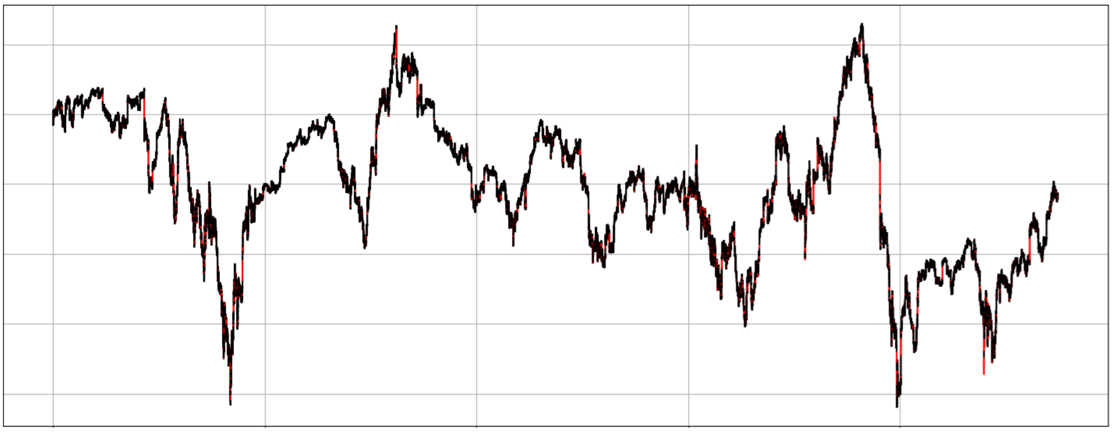


Figure 8.4: QMSE emergency switch for ASM 1min validation set trading.

generalization in high-frequency forecasting contexts and highlights the need for careful evaluation when assessing real-world trading applicability.

It is particularly noteworthy that, given the turbulent market conditions underpinning the evaluation period, it is unsurprising that a 1min forecasting model would have incurred losses during this interval. It can be argued that under such exceptional conditions, the model would not have been used in automated trading. Nonetheless, to quantitatively identify these anomalous periods, an additional experiment was conducted—presented in Figure 8.4—in which the QMSE metrics and d, as proposed in [223], were employed as a emergency-mechanism. Using a threshold parameter, intervals during which trading would have been suspended based on the condition $\theta_{\text{QMSE}} \leq d$ are highlighted in red. θ_{QMSE} must be carefully chosen. If done so, the visualization indicates that several critical periods could have been avoided. It must be emphasized, however, that ASMs are not risk-free models and would have incurred losses during certain market phases. Whether these models—in their current form—are suitable for deployment in live trading is addressed in Section 8.3.

The observation that the sMAPE of most SPP models—especially those discussed in Section 7.7—is much better than that of naive benchmarks, while accuracy against SMP remains relatively low, suggests that these models can approximate future price magnitudes well but struggle with directional prediction. This limitation diminishes their utility as trading models for individual stocks. However, it concurrently implies that forecasting objectives which depend primarily on magnitude rather than direction—such as volatility indices (e.g., VIX 🇺🇸) or other market-wide volatility measures—may represent more suitable targets for these modeling approaches.

Research Question 3

How Can Effective Foundation Models for Quantitative Stock Data be Built?

FIGURE 8.5: Research Question 3 as posed in Section 1.2.

Architecturally, this thesis develops a foundational model for processing quantitative equity data, represented by the proposed ASMs. The requirements of both this thesis and the extant literature (cf. [82]) have been satisfied. The model has been pretrained on a diverse array of heterogeneous datasets (All$^{(2010:)}$) as well as the S&P-500 🇺🇸 dataset. Due to its modular architecture, the proposed model is extensible: additional data types, asset classes, time resolutions, or equities can be added without changing the existing model or trained weights.

The pretraining phase is primarily constrained by the quantity of available training data, as evidenced by the performance disparities observed between the All$^{(2010:)}$ dataset and the higher-frequency S&P-500 🇺🇸 datasets in Section 7.7.1. The applicability of this model to downstream tasks is demonstrated via implementations of SMP and SPP; although these implementations are functional, the literature contains substantially better performing models for these specific tasks. As noted in Chapter 1, the objective of this thesis is to investigate the success of adapting NLP models to quantitative stock data, rather than to develop SOTA SPP or SMP models. In this context, the proposed ASM algorithm constitutes a successful adaptation of LLMs to quantitative stock data.

The suitability of SMP and SPP as downstream tasks for ASMs should be reconsidered based on the overall findings, as the broader predictive usefulness of ASMs remains uncertain. ASMs, as foundational models, exhibit characteristic properties analogous to those of LLMs in NLP: namely, the pretraining plus fine-tuning paradigm can be conceptualized as 'general representation learning → downstream task'. This paradigm manifests in NLP through reduced convergence times on downstream tasks. The main idea is that pretraining performs much of the task-agnostic representation learning—referred to here as 'general language understanding' [220]—so that later fine-tuning requires relatively few updates. Consequently, the ASM framework demonstrates that this two-stage training paradigm, inherent to LLMs, can be successfully transferred to quantitative stock data.

In the ASMs, an analogous approach to the criteria delineated in [62], namely 'learning indicators and interrelationships → time-series indicators and predictions' is posited. Initially, general quantitative representations are learned during pretraining (MPM and the use of pretrained S2V embeddings) prior to any downstream task. The first component of the optimization problem—representation learning—can be clearly demonstrated by the strong pretraining performance, embedding evaluations, S2V model assessments, and model evaluations on SMC and SPE tasks. For the specific objective of time-series forecasting, ASMs prove conditionally useful but do not consistently outperform alternative SPP/SMP architectures. Although pretraining typically yields modest improvements in forecasting performance (as shown in Figure 8.2), these gains are quantitatively minor. Convergence analyses indicate that the core relationships have already been internalized during pretraining, enabling SMP/SPP models to build upon this foundational knowledge. However, given the limited scope for further optimization and the relatively weaker predictive performance of the ASM architecture in forecasting tasks, the value of employing ASMs specifically for SMP and SPP may be questionable.

As in NLP, this architecture is expected to act as a foundational model within larger frameworks and various downstream tasks. Several of these potential applications are described in Section 9.1.

8.3 Limitations of the Research

Key limitations of the research and their potential impact on results and interpretation are outlined in the following.

Dataset Bias and Model Behavior The robustness of the presented results should be critically assessed, especially in terms of their relevance to real-world applications. A key issue, discussed earlier in Section 2.3, concerns the presence of missing (non-moving) data in the stock datasets. A linear interpolation strategy was adopted to mitigate this issue and impute missing values. This decision was motivated by the observation that the datasets themselves and, more notably, the SMP labels exhibit substantial distributional imbalance. This imbalance is particularly pronounced due to the temporal structure inherent in the intraday datasets where $\nexists \theta \in \mathbb{R} : \left|\mathbf{1}^T \cdot \mathbb{I}(X \leq \theta) \cdot \mathbf{1} - 0.5 \cdot \mathbb{T} \cdot |C|\right| \approx 0$ holds true. As a consequence, the model was trained and also evaluated on data that do not occur in this form in reality. This issue is less pronounced in evaluation, where data quality is higher.

The SMP label distribution is partly problematic as non 50/50 distributions must be strongly regulated during training to counteract the Winner and Loser stock behaviors (explained in the following) and $\frac{|\{i|\hat{\mathbf{y}}[i]=0\}|}{|\{i|\hat{\mathbf{y}}[i]=1\}|} \approx 1$ holds true. Separate experiments with loss and accuracy metrics only for the moving element were also carried out.

In particular, simpler models (less pronounced in ASMs) may develop a relatively trivial yet effective strategy for predicting SMP developments. This strategy is referred to as the 'Winner and Loser Stocks' strategy in the following: This strategy is characterized by consistently predicting the same outcome for certain stocks. For instance, so-called "winner stocks" such as BRK.B 🇺🇸 or SEA 🇸🇬 are invariably predicted with $y_{\text{BRK.B}} = 1$ and $y_{\text{SEA}} = 1$, respectively. Conversely, "loser stocks" such as DBK 🇩🇪 or BAYN 🇩🇪 are consistently predicted with $y_{\text{DBK}} = 0$ and $y_{\text{BAYN}} = 0$.

Despite the simplicity of this strategy, this strategy performs well, and simpler models often fail to exceed it. In some cases, the model even exhibits a tendency to assign the same label to all stocks. SF differs from other ML areas, such as NLP, in one relevant respect. Due to the inherent complexity of the task and the anticipated low accuracy, it is possible for 'defective' models — for instance, models with fully exploded gradients or Θ that exhibit severe overfitting and predict only a single label — to still achieve comparatively strong, or even performance levels unattainable by other models.

However, as discussed in Chapter 1, because SF serves as an auxiliary task and the primary goal is to develop a meaningful approach that could be transferred to other areas of ML, certain targeted constraints are introduced in this context. Specifically, the use of models for which $\underline{\overline{\left(\frac{\sum_{i=1}^{|C|} \mathbb{1}(\hat{\mathbf{y}}_i > 0.5)}{\sum_{i=1}^{|C|} \mathbb{1}(\hat{\mathbf{y}}_i \leq 0.5)}\right)}} \gg 1$ holds is restricted, and

$$n((c_i, l)) = \sum_{i=1}^{\beta} \mathbb{1}(\hat{Y}[j][i] = l) \tag{8.1}$$

with

$$P^*((c_i, l)) = \frac{n((c_i, l))}{\sum_{l=0}^{1} n((c_i, l))} \tag{8.2}$$

and

$$\theta > D_{\mathrm{KL}}(P^* || Q) \text{ with } Q((c_i, l)) = \frac{1}{2} \cdot \beta \tag{8.3}$$

(with β as mini batch size) holds, in order to prevent the SMP forecasts from being distributed equally in general, but the most favorable forecast is always made for each stock separately. This happens especially with all models that are not ASMs and mainly without normalization. If the normalization from Appendix A.1 is not used in the ASMs, it can be observed there too.

In both limitations, it is evident that, in many cases, the models could have achieved superior results, particularly with regard to all SMP metrics. In such instances, it is challenging to assess to what extent other models — such as those referenced in Chapter 2 — may have been similarly affected by these phenomena but it was failed to recognize them. Furthermore, it remains difficult to determine whether the models have developed a valid strategy, are exploiting a dataset bias, or merely possess flawed Θ.

Despite these adjustments, certain models still achieve comparatively strong performance with respect to the F1-score. Focusing on the ASM models as foundation models, noteworthy results are observed not only in the SMP and SPP intraday runs (see Table 7.58, Table 7.59, and Table 7.67 and Table 7.51 — particularly in the case of LLaMA) but also in runs where 10 stocks are used as target variables and 60 as input features, as shown in Table 7.54.

Among the 1min runs, BERT, as an encoder-only model, demonstrates exceptionally strong performance both overall and specifically in terms of the F1-score (see Table 7.55). Overall, BERT appears to be particularly well-suited for predictive SMP/SPP tasks, as evidenced across multiple instances in Section 7.7. This is especially true for runs involving a small number of target stocks, as illustrated for example in Table 7.52. Reason for this might be the predictive encoder-only architecture.

The $F^{\langle \text{T} \rangle}$ models demonstrate overall solid, yet partly fluctuating F1-scores across the SMP and SPP tables (e.g., Table 7.33, Table 7.34). Models without additional pretraining phases (e.g., $F^{\langle \text{E-M} \rangle}$, $F^{\langle \text{J-M} \rangle}$, $F^{\langle \text{J-C} \rangle}$) achieve decent accuracy scores. The SPP models are usually much better in terms of F1-score (at least in intraday settings and especially in the ASMs). Pretrained variants (e.g., $F^{\langle \text{E-M} \rangle} + \mathcal{L}_p$ or $F^{\langle \text{J-M} \rangle} + \mathcal{L}_p$ in the SPP tables) are capable of partially stabilizing class balance, yet the effect is very small. In SMP the pretraining has an effect on the F1-score, especially in the extreme cases (e.g., $F^{\langle \text{E-M} \rangle}$ in the SMP 60min setting, cf. Table 7.34 and Table 7.40, or in the SMP 1min setting, see Table 7.35 and Table 7.41).

In direct comparison, the ASM models generally produce more balanced F1-scores (except for a few outlier models as for example the S2V embeddings in Section 7.7.2 and Section 7.7.3) and are therefore less prone to biased predictions. However, $F^{\langle \text{T} \rangle}$ models can reach similarly high F1-scores and remain competitive, especially at higher frequencies. In particular, the unpretrained $F^{\langle \text{J-C} \rangle}$ model achieves the highest F1-scores among all transformer variants in the SMP interday setting (0.505/0.517, see Table 7.33). Similarly, the unpretrained $F^{\langle \text{J-M} \rangle}$ model in the SPP interday task stands out with F1-scores of up to 0.531 (see Table 7.36).

It should be emphasized once again that significantly better results could have been achieved across all datasets (in terms of accuracy, F1-score, and sMAPE) if the phenomena mentioned at the beginning of this section had been ignored. However, refraining from doing so was deliberate in order to maintain the meaningfulness of the investigation and avoid obtaining trivial models. Further information can be found in Appendix A.5.

Data Volume Limitations The data limitations of stock market data are frequently discussed in academic literature (see Chapter 2 and especially Section 3.0.4). Given the limited availability of historical data, the relative youth of modern stock markets, and the lower recording frequency of interday data points, direct comparisons between interday and intraday data are currently not feasible. It would be useful to test whether performance differences stem from lower market efficiency at 1min or simply from larger data volume. Experiments exploring these aspects will become feasible only in the distant future.

Experiments have also been conducted with 22 ETFs from 12 countries (see Appendix A.2) for all models in this thesis, but found that the shorter time periods that have to be used, since the first data for the ETFs are mostly in the late 2010s, affect the training of the data-hungry models worse than the marco economic information of the ETFs can compensate.

The observed underperformance of the $\text{All}^{(2010:)}$ dataset in the MPM of the ASMs can plausibly be attributed to the limited data volume, as evidenced by the outcomes observed in the 1min and 60min runs of the S&P-500 🇺🇸 compared with the S&P-500 🇺🇸 interday runs. A possible explanation for this phenomenon is provided in [177].

In [177], a GNN is constructed under the premise that stocks DWT or similar methods, along with the application of individual threshold values to define the edges in the adjacency matrix for the GNN between stocks, is not advisable. The authors argue that the number of edges adheres to a power-law distribution in node degree, as described in [198], which becomes more prominent with increasing $|C|$.

Although ASMs lack explicit edges, it is assumed in the subsequent discussion that an ASM with comparable capabilities to a GNN in representing inter-stock correlations would similarly increase in complexity as the number of edges in the corresponding GNN grows. Furthermore, [177] substantiates that the number of nodes associated with a given stock scales according to a Zipf distribution, specifically with complexity $\mathcal{O}(k \cdot |C| \cdot \log(|C|))$, implying that not every stock must be connected to every other stock, thereby circumventing the complexity of $\mathcal{O}(|C|^2)$.

Building on this premise, representing interrelationships in the MPM remains a substantial challenge, as the complexity continues to scale by a factor of $k \cdot \log(|C|)$ as $|C|$ increases, with all such relationships still requiring representation in the model.

Given the limited training data available for the All$^{(2010:)}$ dataset, as detailed in Chapter 4, the model likely suffers from significant underfitting in this regard. This is particularly pronounced because the model must not only account for individual stocks but also represent all corresponding relationships. This observation is further supported by the experimental results in Section 7.7.1, where setting $\zeta = 100$ clearly improved performance.

One effective strategy to address data scarcity involves pretraining on larger datasets, such like those containing 1min intraday data. Similar to LLMs, which perform well on small fine-tuning datasets due to their extensive prior knowledge, time series models also benefit from this approach. TransformerXL and GPT-2 exhibit clear improvements in performance when pretrained on high-frequency data / higher data volumes, as illustrated in Table 7.59, whereas LLaMA produces less consistent results. Notably, GPT-2 shows increased sensitivity to hyperparameter configurations despite sometimes converging early, as evidenced in Table 7.51. As can be seen in Table 7.59 for SMP, pretraining on 1min intervals usually results in a deterioration in performance (these often do not achieve baseline performance and are therefore not tabulated). This may be attributed to the optimization process already being advanced (including wrt. downstream tasks), leaving little room for improvement and causing rapid overfitting. This would indicate that models exhibiting very strong MPM performance are primarily affected.

Furthermore, the stabilizing effect of pretraining on both MCC and F1-scores is confirmed in the 60min setup (Table 7.60). However, performance gains are not uniform and vary depending on the model backbone and the pretraining frequency (1min vs. 60min data). Both the high-volume 1min pretraining (see Table 7.63) and the more temporally aligned 60min pretraining offer a measurable advantage over models trained from scratch, as indicated by improved final accuracies and reduced standard deviations.

Application Settings and Barriers In practical applications, predictive models for forecasting stock prices exhibit several inherent limitations that may constrain their applicability and profitability beyond theoretical validation. First, real-world transaction costs—including order fees, bid-ask spreads, and slippage—reduce gross returns but are often insufficiently considered in academic backtests (as in [66]). This issue is particularly pronounced in high-frequency trading strategies, where cumulative costs may entirely offset model-generated profits.

Second, even with high predictive accuracy, models can experience periodic drawdowns or long loss periods, which may discourage both institutional and private investors. Such phases weaken the risk–return profile and reduce confidence in the model's stability.

Third, inference time constitutes a critical bottleneck: In latency-sensitive domains such as high-frequency trading, even millisecond-level delays may render predicted signals obsolete before they can be executed. Recent deployment notes stress model compression (distillation/quantization), domain-adaptive prefix-tuning, explicit slippage/impact modeling, and Order Management System/Market Risk Management integration as prerequisites for production use [289].

Fourth, data quality emerges as a pivotal factor. While backtesting typically relies on cleaned and fully labeled datasets, real-time market data often contain missing values, inconsistencies, and anomalies. These issues can significantly impair prediction accuracy. Importantly, this challenge is not mitigated at lower trading frequencies: even here, elevated inference latency can lead to a temporal mismatch between signal generation and order execution, resulting in missed alpha opportunities.

Furthermore, as shown in Section 7.7.5, the model exhibits a poor MDD, implying a generally high level of risk exposure.

These limitations align with recent survey findings highlighting the open challenges of data reliability, regulatory considerations, interpretability, and out-of-sample validation for LLM-driven equity research [287]. Real-world deployment of otherwise robust SF models is constrained by operational, technical, and data limitations.

8.4 Ethical Considerations

All experiments were conducted strictly for research and demonstration purposes. No live trading interfaces were deployed, no automated orders were placed, and no market interventions were performed. Any discussion of potential applications is exploratory and must not be construed as financial advice. The work uses only offline evaluation and simulated scenarios.

In line with the intent of IEEE Std 7001 on transparency of autonomous and intelligent systems ,[283], this work follows the principle of stakeholder-appropriate and testable disclosures. Data statements regarding sources, representativeness, and biases are documented in Chapter 5, while model cards summarizing purpose, training setup, evaluation metrics, and limitations are reflected throughout Section 4.2 and Section 5.2. Preprocessing steps, evaluation windows, and metrics are consistently reported in the methodology chapters to support auditability and replication. Taken together, these measures align with 7001's emphasis on evidencable transparency across the system lifecycle.

Following the lifecycle guidance of IEEE Std 7003 on algorithmic bias considerations ,[284], this study establishes a bias profile through the reporting of class distributions and dataset pathologies in Section 5.2, the enforcement of strict temporal splits in Section 4.2, and robustness checks across different market regimes in Section 4.3. Mitigation strategies such as thresholding and calibration are discussed within the experimental context, while the limitations section (Section 8.3) reflects on residual biases and dataset constraints. Since no deployment was performed, monitoring requirements remain out of scope; nevertheless, the prerequisites for future deployment are outlined in accordance with the 7003 standard.

No new human-subjects data were collected. The research relies on publicly available and/or contractually licensed market data and, where applicable, publicly available text subject to the respective terms of use. Personally identifiable information was neither intentionally processed nor inferred, and features were aggregated at the instrument or market level. Proprietary third-party datasets are not redistributed and are used solely under license. To enhance transparency and reproducibility, the project maintains concise data statements describing sources, selection rationale, representativeness, known biases, and usage rights, and model cards summarizing purpose, evaluation conditions, limitations, and appropriate use.

Financial data exhibit non-stationarity, regime shifts, and sectoral or regional imbalances. To mitigate methodological artifacts, the experimental design enforces strict temporal separations between training, validation, and test sets, employs robust baselines, and evaluates across multiple market phases. Checkpoints are reused when feasible, and efficient training schedules are favored to reduce the environmental footprint without compromising scientific validity.

This dissertation is a research artifact and was not deployed in regulated production settings. Any real-world deployment would require adherence to applicable regulatory frameworks, independent validation, comprehensive backtesting across regimes, stress testing, ongoing monitoring, and governance consistent with established model risk management principles. Predictive models and market representations can be misused, and although this work omits deployable trading agents, real-time signal endpoints, and tooling intended to influence prices, the presented models and techniques are sufficiently foundational and general that their usefulness to abusive applications cannot be categorically ruled out.

Chapter 9

Future Work and Conclusion

This chapter outlines avenues for future research and reflects on the broader implications of the presented findings. Building on the contributions of this dissertation, it highlights potential applications of ASMs beyond the core experiments, discusses fine-tuning scenarios in financial time series analysis, and concludes with an overall assessment of the work.

9.1 Alternative Use Cases as Future Finetuning Tasks

This section discusses a range of potential downstream applications for ASMs. These use cases illustrate how pretrained models can be adapted through fine-tuning to address domain-specific objectives in financial time series analysis.

Overview of Alternative Use Cases Pretrained LLMs in NLP undergo rigorous evaluation across an extensive spectrum of downstream tasks, with novel benchmark challenges continuously emerging to assess their generalization capabilities. As articulated in [82], the foundational model should encapsulate this generalized market understanding. Consequently, all subsequent use cases should be conceptualized as (domain-specific) fine-tuning tasks.

A concept introduced in the authors prior work [223] involves the application of ASMs over short temporal windows for the identification of rapid market disruptions, such as flash crashes, and the detection of anomalous market conditions. This aligns with the conceptual framework proposed in [278], where Time Series Anomaly Detection is discussed as a potential pretraining objective within the TSFM models. Furthermore, [278] explores additional applications, including time series classification, both long- and short-term forecasting, as well as Few-Shot and Zero-Shot forecasting. The latter can be realized either implicitly through domain shifts in stock market data or as an explicitly defined task.

Moreover, potential extensions of this framework include applications in risk-aware capital management and portfolio optimization (see Section 9.1), alongside the SDM task (see Section 9.1) or the Lead-Lag Strategy as introduced in [82], which revolves around trading two assets where the price movements of a ‘leading’ asset are expected to influence those of a ‘lagging’ asset.

ASM Based Market Simulations Given the robust performance observed in the MPM task, it can be inferred that an ASM achieves high classification accuracy when provided with a market C and the corresponding sliding window X, under the condition that two disjoint substructures $\dot{X}$ and $\ddot{X}$ exist such that $\dot{X}, \ddot{X} \trianglelefteq X$ and $\dot{C} \cap \ddot{C} = \emptyset$ holds and one of these windows is known i.e. estimated. As an initial configuration, $\dot{C}$ may be defined as the subset of stocks within which a (potentially high-risk) investment is intended, constituting the portfolio of interest. Conversely, $\ddot{C}$ represents the subset of stocks for which there exists either a strong certainty regarding a future movement or a concern regarding a potential future movement. A reliable method for estimating these subsets with enough precision to reduce risk is needed. Concretely, in ASM based MPM models $\mathcal{P}_{\ddot{X}^{(t+\omega)}}(\mathcal{X} = \mathbb{1}^{(t)}(\dot{X}^{(t)} > \dot{X}^{(t+\omega)}))$ is computed.

Furthermore, the modular and extensible architecture of ASMs allows $\dot{C} \cup \ddot{C} \neq C$. This property is especially relevant in practice, as it recognizes the impossibility of accurately predicting the dynamics of the entire market C, while allowing a focused analysis of sector-specific effects. Such a structural flexibility is advantageous, as market participants are frequently more concerned with the behavior of a specific industry rather than the aggregate market. This assertion is further substantiated by the (relatively) strong empirical performance demonstrated by the MPM models using the ζ-approach, as discussed in Section 7.7.1.

Beyond the selection of an appropriate subset $\ddot{C}$, the most critical aspect is the model's capability to generalize to the SMP labels, specifically

$$\mathcal{P}_{\left\{\mathbb{1}^{(t)}(\ddot{X}^{(t-j)}>\ddot{X}^{((t-j)+\omega)})\right\}_{j=0}^{\Delta t}}\left(\mathcal{X}=\mathbb{1}^{(t)}(\dot{X}^{(t)}>\dot{X}^{(t+\omega)})\right) \tag{9.1}$$

, which has to be tested in future research. Alternatively, the tolerance for $\mathcal{P}_{\ddot{X}[i,j]+\epsilon_{i,j}}(\mathcal{X}=\mathbb{1}^{(t)}(\dot{X}^{(t)}>\dot{X}^{(t+\omega)}))$ should be examined, i.e. if regressive values are estimated but they deviate from the real, exact future values and are therefore noisy.

This formulation is particularly relevant, as estimating directional movements is inherently more feasible than predicting precise regression values. A secondary yet equally significant consideration is the sensitivity of the model's performance to variations in ν_M and Δt. Specifically, determining the upper and lower bounds for these parameters while maintaining predictive efficacy is crucial, as it directly impacts the number of data points required for estimation.

Formally, this challenge can be framed as the optimization problem

$$\min_{\Delta t} \quad \max_{\nu_M} \quad \nu_M^*, \quad \Delta t^* \quad \text{s.t.:} \quad \mathcal{L} \leq \mathcal{L}^* \tag{9.2}$$

for future research.

ASM based Risk Modeling and Portfolio Optimization Additionally, a prospective portfolio optimization algorithm inspired by the foundational principles of Markowitz [153] is proposed. As demonstrated in [183], ML models can be effectively utilized for portfolio optimization. The general approach works by imposing either a predefined minimum expected return μ or an upper bound on risk ς. In this context, these parameters are redefined independently from their original mathematical formulations in [183] to align with the specific modeling framework. Each stock $c_i \in C$ is assigned a corresponding portfolio weight w_i, collectively represented as the weight vector $\mathbf{w} \in \mathbb{R}^{|C|}$. The portfolio weights are subject to the standard constraint $\mathbf{1}^T\mathbf{w} = 1$.

The expected return μ is derived from the predicted relative returns. Notably, absolute returns are abstracted from, and both long and short positions are inherently considered. Formally, $\mu = \mathbf{w}^T \mathrm{abs}(\hat{\mathbf{y}})$ is defined, where $\hat{\mathbf{y}}$ can be generated by any arbitrary ML model—including ASMs—or derived from human-based forecasting. Now ς is redefined accordingly. Instead of employing the covariance of returns as in the original Markowitz formulation, the risk is expressed in terms of two ASM-based components as

$$\varsigma = \underbrace{\lambda_e \cdot \sum_{i=0}^{|C|} \sum_{j=i+1}^{|C|} w_i \cdot w_j \cdot \frac{\mathbf{e}_i^T \cdot \mathbf{e}_j}{\parallel \mathbf{e}_i \parallel \cdot \parallel \mathbf{e}_j \parallel}}_{1)} + \underbrace{\lambda_s \cdot s}_{2)} \quad . \tag{9.3}$$

The first term 1) represents an enhanced, ASM-based formulation of stock dependencies, where $\mathbf{e}_i$ can be defined either as $E[i]$ or as its scaled variant utilizing $F^{\langle \text{FEW} \rangle}$, derived from predicted (or estimated) or historical returns. This approach provides a significantly more granular representation of complex variances—particularly within a temporal context—compared with conventional covariance-based risk measures.

The scalar s in 2) corresponds to the risk output generated by the MPM method. This risk measure is defined as $\mathbf{1}^T\mathbf{s} = s$, and $\mathbf{s}[i] = -\mathbf{y}[i] \cdot \tanh(\hat{r}_i)$ with $F^{\langle \text{ASM} \rangle}_{\mathbf{y}_{\forall j \neq i}} : X \mapsto \hat{r}_i$. In the vocabulary-based approach, the logits associated with the position for stock c_i can be directly utilized to obtain a confidence measure. An alternative approach would involve defining a confidence coefficient that quantifies the minimum proportion of correct predictions required to maintain statistical reliability.

Doc2Vec for Risk Management As previously outlined in [223], QMSEs exhibit significant potential for applications in both risk management and portfolio optimization. In this context, instead of employing QMSE-based distance metrics d as a regularization term in the learning process for $\mathcal{L}_{\text{q-reg}}$—as discussed in Section 6.7.2—these metrics can be leveraged to quantify the risk associated with a given temporal epoch within a portfolio optimization framework.

To accommodate this approach, the constraint $\mathbf{1}^T\mathbf{w} = 1$ may be relaxed, analogous to the extensions of the Black-Litterman model introduced in [155]. This relaxation allows for the allocation of a higher proportion of capital to cash holdings during periods of elevated market risk (i.e. $d \propto (\mathbf{1}^T\mathbf{w})^{-1}$), rather than committing it to investments.

Alternatively, this risk-aware portfolio adjustment strategy can be implemented through trading simulations, similar to those employed in [54] and [66] for SF. These simulation-based approaches offer an empirical framework for evaluating the impact of QMSE-driven risk measures on portfolio allocation strategies.

SDM The task of SDM is proposed as a downstream task, a relatively underexplored approach within the financial forecasting domain. Rather than attempting to predict precise stock values, this methodology focuses on determining a probable value range for stock movements. This approach is rarely explored in the existing literature. A notable implementation of this strategy can be found in [157], which employs distribution predictions across a diverse portfolio comprising 31 assets, including 9 ETFs. SDM can be implemented as an initial fine-tuning step by extending the $F^{\langle \text{CLS} \rangle}$ weight matrix to match the parameters required by the chosen distributional approach. In the case of Gaussian modeling, for which preliminary experiments have already been done for this thesis, this can be realized by defining $W_{\text{CLS}} \in \mathbb{R}^{\xi \times 2 \cdot |C|}$. Here, the first partition of $\hat{\mathbf{y}}$ corresponds to $\vec{\mu}$, whereas the second partition is allocated to $\vec{\sigma}$.

Analogously, the Weibull distribution—parameterized via $W_{\text{CLS}} \in \mathbb{R}^{\xi \times 3 \cdot |C|}$ to accommodate λ, θ, k—or the gamma distribution, which necessitates the parameters k and θ, can be effectively incorporated. These distributions enable the modeling of directional tendencies concerning one side of the actual value x_i^t, thereby providing an estimate of movement confidence.

Such an approach is also applicable to risk assessment. Using the Gaussian distribution as an illustrative example, the directional position (long or short) can be inferred based on $\vec{\mu}_i$. Risk exposure can be quantified by classifying predictions with most probability mass on the loss side as high risk. This risk measure can be formally expressed as the proportion of the distribution area residing on the 'incorrect' side as

$$w_i \propto \left(\int_{\vec{\mu}_i^t}^{\vec{\mu}_i + 3 \cdot \vec{\sigma}_i} \varphi_{\vec{\mu}_i, \vec{\sigma}_i}(a) da + 1 \right)^{-\alpha} \text{ or } w_i \propto \left(\int_{\vec{\mu}_i - 3 \cdot \vec{\sigma}_i}^{\vec{\mu}_i^t} \varphi_{\vec{\mu}_i, \vec{\sigma}_i}(a) da + 1 \right)^{-\alpha} . \tag{9.4}$$

Alternatively, the expected return can be redefined as

$$\hat{\mathbf{y}}_i \leftarrow \int_{\vec{\mu}_i - 3 \cdot \vec{\sigma}_i}^{\vec{\mu}_i + 3 \cdot \vec{\sigma}_i} \varphi_{\vec{\mu}_i, \vec{\sigma}_i}(a) \cdot (\vec{\mu}_i - a \cdot \operatorname{sign}(\vec{\mu}_i) - \vec{\mu}_i \cdot (1 - |\operatorname{sign}(\vec{\mu}_i)|)) da \tag{9.5}$$

using the Gaussian distribution as an example again.

Decoder Approach for ASMs As outlined in [224], various methodologies for leveraging the generative capabilities of most LLMs have been proposed. However, in contrast to other research approaches, the intention is to continue utilizing these models for predictive forecasting (or for generative predictions with a shortly constrained temporal horizon). This decision is motivated by the considerations discussed in Chapter 2, particularly the observation that even short-term predictions with $\omega = 1$ exhibit suboptimal performance. Notably, [110] addresses the rationale for multi-day predictions by noting the regulatory requirements imposed on institutional investors. Specifically, it states that financial regulators require a liquidity horizon of at least ten days for institutional investors to sell risky stocks, a rule meant to prevent major market price disruptions.

An alternative approach is proposed in which the model, when generating predictions for short forecasting horizons, autonomously determines for which c_i predictions should be made, thereby selecting instances where it exhibits higher confidence. Formally, the decoder is defined as $F_\rho^{\langle \mathrm{D} \rangle} \mapsto \tilde{B} \in \mathbb{R}^{\xi \times l}$. In the context of LLMs, $\tilde{B}$ is typically processed through a linear transformation with a weight matrix of dimensions $\tilde{\xi} \times |\tilde{V}|$, followed by a softmax activation. During training, each component of the output is mapped to a corresponding word token using the cross-entropy loss function ($f_{\text{Cross-Entropy}}$), whereas in autoregressive inference, token selection is performed via (multinomial) sampling based on the computed logits.

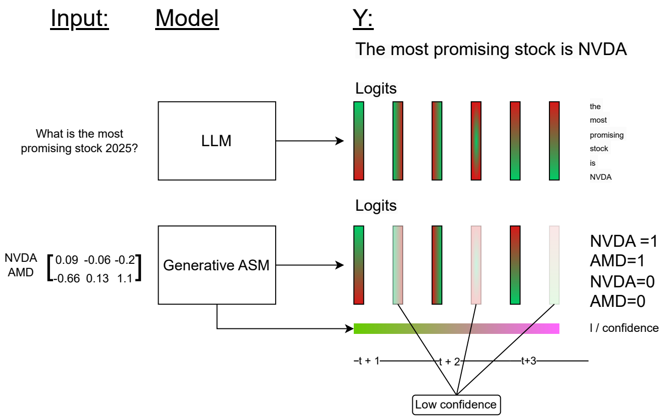


FIGURE 9.1: Sketch of the proposed generative approach.

For the SF models, this framework is extended by employing vocabulary-based models, wherein the linear projection layer $F^{\langle \mathrm{G} \rangle}$ is defined with a weight matrix $W_G \in \mathbb{R}^{\xi \times (|C| \cdot 2+2)}$. This formulation accounts for each stock and movement individually while additionally incorporating an [EOS] token and a [PUNC] token. A straightforward approach could involve defining the target variable as $Y \in \{0,1\}^{(\theta+1) \times (|C| \cdot 2+2)}$, initialized as a zero matrix with the exception of

$$Y\left[i, f\left(X_{i \bmod |C|}^{(t+1+([\frac{i}{|C|}]))}\right)\right] = 1 \tag{9.6}$$

where $f(x_i^{(t)}) = \mathbb{1}^{(t)}(x_i^{(t)} > x_i^{(t+1)}) \cdot |C| + i$, with the final position reserved for the [EOS] token.

Additionally, [PUNC] tokens may be interspersed between different time steps t, though empirical evaluations indicate that their inclusion does not yield notable benefits. Consequently, for $\omega = 1$, this approach offers no conceptual advantage over the previous methodology used in this thesis. However, for $\omega = 2, 3, \dots, \theta$ it facilitates a generative modeling strategy that incorporates an extended historical horizon. This formulation closely parallels the structure of conventional generative text generation tasks in LLMs, as illustrated in Figure 9.1.

As introduced in [224] an alternative and potentially more effective approach involves allowing the model to autonomously determine for which stocks predictions should be generated, incorporating only these into the loss calculation. To prevent the model from consistently producing excessively short or minimal predictions, non-selected instances could either be excluded from the loss function or penalized with a minimal regularization term.

Initially, methods were explored in which the model was granted complete freedom to generate a token at each position while simultaneously self-assigning token representations corresponding to specific company predictions. However, this approach resulted in highly unstable model behavior.

A more robust strategy is to compute predictions over a time horizon θ, where the total sequence length is given by $l = |C| \cdot \theta$. Each company c_i and time step t is then assigned a fixed position within this sequence. For SMP, the vocabulary-based approach is subsequently employed while also computing $\hat{\mathbf{I}} \in (0, 1)^l$ with σ activation, which quantifies the confidence level associated with each prediction. Subsequently,

$$\mathcal{L}_{\text{Gen}} = \frac{1}{l} \cdot f_{\text{Cross-Entropy}}(\hat{Y}[i], Y[i]) \cdot \hat{\mathbf{I}}[i] + \lambda_l \cdot (1 + \mathbf{1}^T \cdot \hat{\mathbf{I}})^{-1} \tag{9.7}$$

is defined, which enables the model to assign lower confidence to specific points either within the loss function or its predictions, thereby allowing for uncertainty estimation. To prevent the model from systematically applying this mechanism to all predictions, a posterior regularization term enforces a constraint that encourages the overall confidence to remain as high as possible.

Experiments with this method have been done on interday data. A major challenge lies in the exceptionally high standard deviation in performance, which can be attributed to the inherent current instability of the approach. Another issue is that the model sometimes assigns very different confidence values across runs (which is desirable), while in other cases, the values stay almost uniform. This inconsistency should be examined in future work. Moreover, determining an appropriate value for λ_l is particularly challenging, as it strongly depends on the current selection of C.

However, when a stable model is obtained, the predictive performance across multiple future time steps is promising. Specifically, for the T5-based model, SMP accuracies ranging between 50.5% and 51.0% can be achieved. Furthermore, the model's predictive stability can be maintained for up to 13 trading days ahead.

9.2 Further Future Work

Future work should focus on ASMs as foundation models, reflecting their role as the main contribution of this dissertation. Nevertheless, numerous underlying concepts and, in particular, the downstream applications are transferable to the other methodological frameworks proposed in this thesis. The proposed avenues for future investigations include; ideas derived from established literature, the expansion of (pretraining) datasets, the incorporation of fundamental (T/TST) data sources (e.g., multimodal models in NLP), and the adaptation of the ASM methodology to broader classes of multivariate time series domains. Accordingly, this section is structured into four distinct subsections, each dedicated to one of these key aspects. The exploration of additional fine-tuning tasks and further use cases of the foundation models have been already discussed in Section 9.1.

9.2.1 Specific Future Model Investigations

Several potential enhancements to the models discussed in this dissertation—identified in the literature but not explored in detail—warrant further investigation. The approach introduced in [134] advocates for the utilization of multiple prediction heads, dynamically switching between them based on identifiable trading patterns. This methodological refinement is broadly applicable to various model architectures and, in particular, could be effectively integrated with the dedicated prediction heads $W_{\text{ASM-TP}}$ per c_i as outlined in Section 6.11.2.

A conceptually related approach was introduced in [213], where the authors argue that, within investment funds, multiple experts contribute distinct insights before a final decision is reached. This decision-making process is subsequently mirrored in ML pipelines through the integration of diversified or differently initialized models. A similar strategy could be employed in the models presented in this thesis.

Furthermore, the adaptive strategy proposed in [130], which involves categorizing market conditions into extreme and normal states and subsequently employing distinct model components for each scenario, represents another promising direction for future model improvements. Doc2Vec models could serve as identification mechanisms for these states within the proposed framework.

The normalization approach proposed in [141] was incorporated only to a limited extent, leaving room for a more comprehensive examination of de-stationary attention mechanisms in future research endeavors. Additionally, [168] introduces alternative normalization techniques specifically tailored to stock market models, which dynamically adjust to the underlying data. Initial findings regarding their practical implementation are also discussed therein. Findings on latent-noise models [84] and critiques of deterministic SF [33] indicate useful directions for further study. Specifically, this includes the exploration of SDM as a fine-tuning task, as elaborated in Section 9.1, as well as the increasing relevance of non-deterministic modeling approaches in light of advancements in quantum computing.

To further enhance the performance of the TM task or to develop time series embeddings that exhibit sensitivity to subsequent trends for executing NSP task adaptions, the ideas introduced in [223], which builds upon the framework of [264], could be further refined. The idea here is an approach that adopts an alternative perspective by initially generating embeddings in which future price trends function as implicit labels. Although these labels are not directly used for prediction, they are essential for shaping the spatial arrangement of embeddings in the vector space, where the Frobenius norm is the main metric used to enforce this structure. Furthermore, the Frobenius norm may also be leveraged in alternative configurations within TM, particularly by ensuring that temporally adjacent time series maintain close proximity within the TM vector space. Alternative approaches were presented in [48] for individual assets, but can possibly be adapted for whole stock trends.

In the majority of the experiments, the primary focus was on the OHCLV feature set, with the exception of the technical indicators discussed in Section 6.11.2. However, expanding the feature space to incorporate additional indicators, such as the RSI or SMA, as well as fundamental data that extend beyond the OHLCV representation, could prove beneficial for improving model performance.

Tokenization remains a promising direction for future work. This method, with digit-level tokenization, offers the potential to enhance the numerical stability of models in the context of non-stationary time series data. Potential refinements include summing embedding vectors to reduce input sequence length, exploring alternative tokenization strategies, or leveraging different LLMs or processing architectures. In Chapter 2, several approaches that embed time series data into LLMs have already been outlined, offering a foundation for further exploration in this direction.

Meta-learning remains a promising direction and has been explored in [264] [150].

9.2.2 Diversifying the ASM Data

Building upon the methodologies proposed in [82] and [45], the integration of more data sources and perspectives—from 'macro (e.g., markets, policies, economy), meso (e.g., industries), to micro (e.g., stocks, companies)' [45] levels as outlined in [45]—could enhance the model's performance. A feasible approach to achieving this expansion involves the pretraining of the ASM on additional datasets. This is facilitated by the model's intrinsic scalability and adaptability. However, prior to such an extension, it is imperative to resolve the challenge discussed in Section 7.7 and systematically evaluate the impact of incorporating an expanded set of assets within C while ensuring the preservation of high pretraining task performance. The extended dataset may encompass a broader spectrum of financial instruments, asset classes, markets, and temporal intervals.

During fine-tuning, tasks such as high-frequency trading, LOB processing, and long-term market trend prediction should be assessed to determine where ASMs perform better or worse. The availability of extensive datasets would enable further in-depth investigations, facilitating a more granular understanding of the model's applicability across diverse financial domains.

9.2.3 Fundamental Data

The incorporation of fundamental data, as Π can be done in three distinct ways.

Firstly, quantitative ASMs may be integrated into a multimodal framework, analogous to their application in NLP, where numerous LLMs serve as components of V+L models. This approach can be extended to the models outlined in Section 2.3 by using the ASMs to process the TS component within the architecture or its pipeline. An alternative approach entails representing fundamental data across diverse types and modalities in the form of embeddings, as $\mathbf{e}_{\Pi}$, which are incorporated as $A^{(t)} \odot \Pi$ (as defined in Section 6.11.2). Furthermore, $\Pi = F^{\langle \mathrm{E} \rangle}(f(X))$, derived from Section 6.7.1, remains applicable in this context, as supported by [223].

Additional Π Information Vector Integration in the ASMs Due to the modular and expandable architecture of ASMs, the integration of diverse data sources into the model is theoretically feasible. This includes T data, where ASMs can function as TST models, as well as other numerical fundamental variables such as interest rates and currency exchange rates. Consequently Π enables the ASM to approximate the properties of CLM while operating as an adapated language model.

Furthermore, T data extracted from sources such as social media or financial reports can be incorporated through end-to-end NLP pipelines, e.g. using the model from [221]. A practical advantage of the ASM structure is its temporal integration: textual information is included only when relevant events (e.g., a tweet) occur, rather than at every time step. In experiments, the dataset from [221] and the ACL-18 dataset were processed with FinBERT, and the resulting embeddings were fed end-to-end into a T5-based ASM. However, the observed performance gains were marginal. This outcome is likely attributable to the limited size of the dataset and its relatively short historical coverage. Future research should further investigate these limitations and explore potential enhancements to the methodology.

Building upon the methodologies proposed in [4] and [88], an alternative approach involves the integration of heterogeneous quantitative data across multiple temporal resolutions within time series models. Specifically, data sampled at different

intervals—such as interday (e.g., previous-day values), 60min, and 1min frequencies—can be simultaneously incorporated, with the model receiving explicit embeddings that encode the temporal granularity of each input (serving as Π).
Moreover, following the framework outlined in [88], structured numerical data, including exchange rates, futures contract prices, and ETF prices, can be periodically introduced into the model to enhance its predictive capabilities.

Adapting V+L Models As previously discussed in [220], fundamental models/TST models, provide a suitable conceptual model for comparison with V+L models within the domain of NLP. The integration of NLP with other modalities is an active area of current research. One of the most widely studied approaches is the fusion of NLP with visual modalities.

In the context of this doctoral research, stock market data is conceptualized as a linguistic modality and subsequently processed using NLP models. Accordingly, textual inputs in these models can be replaced with stock data to form a text-to-image (T2I) setup, where the image represents the secondary modality in V+L models. Conversely, V+L models are inherently designed to process textual data within the textual modality. Accordingly, text processing can be retained in the textual stream while stock market data are added as a second modality. This approach can be conceptualized as Stock-to-Image (S2I), wherein stock data is interpreted as an additional modality analogous to the image component in conventional V+L models.

The feasibility of utilizing image-processing architectures for stock data analysis has been previously demonstrated in [74]. In the initial experiments, the X-VLM [262] model from Zeng, Zhang, and Li is being adapted. However, future research endeavors should prioritize the exploration of additional models to further advance the field. X-VLM is one model from the series of transformer-based V+L models that have a similar structure. A textual stream is built with transformers and a visual stream is also built with transformers which are then merged in another transformer-based section to execute a downstream task.

To establish an initial empirical foundation, a series of preliminary experiments utilizing interday stock data are proposed, leveraging the dataset introduced in

[221]. For the textual stream, both the original textual encoders from the respective models and FinBERT, as proposed in [252], are employed. Additionally, a version of FinBERT, as described in [221], is incorporated.

In the Text-to-Image (T2I) paradigm, the image representation is derived from the pretrained BERT model introduced in [221] which processes textual data as input. To integrate stock market data into this framework, the stock data undergoes a linear transformation to an embedding dimension of $\xi = 512$, ensuring compatibility with the model architecture. Subsequently, the outputs from both streams are incorporated into the multimodal model, where the textual stream serves as the query, while the stock data is utilized as keys and values.

There exist multiple methodologies for employing stock data X as the image-equivalent modality in the visual stream. One proposed approach involves leveraging a pretrained ASM, such as the T5-based one. An alternative approach involves utilizing image processing models for stock data representation. X-VLM incorporates CLIP and Swin transformer implementations for visual processing. While these models are originally designed for image analysis, prior research has demonstrated their applicability to stock data, as shown in [74].

To adapt these models for financial data, $|C|$ is reduced, and $\Delta t = |C| \cdot \mathbb{F}$ is ensured, thereby constructing a quadratic image representation. Ideally, both dimensions should be e.g. 384 to integrate with the standard Swin transformer architecture. The Swin transformer implementation is capable of processing an arbitrary number of input channels, denoted as $\mathbb{c}$, which is particularly advantageous when incorporating $\hat{X} \in \mathbb{R}^{|C| \times \Delta t \times \mathbb{F}}$ instead of the original stock data representation X. To achieve this, the number of channels is set to $\mathbb{c} = \mathbb{F} = 5$, thereby encoding each feature as a distinct channel within an image representation. Additionally, experiments were conducted on generating concise summaries of the extended textual descriptions present in the AV dataset and other financial datasets. These experiments utilized state-of-the-art summarization models, such as those proposed in [117] and [140]. While an end-to-end summarization approach could be implemented, this direction has not been pursued further in the current study.

Additional experiments were conducted utilizing the ACL-18 dataset to further assess the efficacy of the proposed models. Empirical results indicate that the incorporation of T data can enhance interday performance to up to 55%, representing a significant improvement over the models from this thesis. However, when compared to alternative models trained on other ACL datasets, this performance remains suboptimal. A fundamental limitation of SMP/SPP based on textual data lies in the inherent dependence of predictive accuracy on the degree to which individual textual inputs align with actual stock movements [54]. Despite these challenges, the integration of textual data constitutes a promising avenue for future research. The potential for further advancements lies in three primary directions: (i) the continued exploration of V+L and multimodal architectures presented in this work, (ii) the refinement and extension of the novel methodologies introduced herein, and (iii) leveraging the modular extensibility of ASM frameworks, as discussed in the preceding section, by incorporating textual data as additional vector representations $\mathbf{e}_\Pi$ within $\mathcal{A}^{(t)}$.

9.2.4 Generalizing NLP Strategies for Multivariate Time Series Prediction

The idea of deriving general frameworks and strategies from NLP—specifically transformers—and adapting them for domain-independent problems such as multivariate time series was first introduced by Zerveas et al. [263]. As outlined in Chapter 2, multiple studies have adopted methodologies inspired by [165] to address the challenges associated with multivariate time series modeling. Efforts to develop universally applicable methods that generalize across various multivariate, time-dependent prediction tasks—and, to some degree, across different domains—have been discussed in [278]. This work specifically references the patching strategies introduced in [165] for the structured processing of c_i. These approaches leverage non-euclidean structures like relationship graphs, spatial structures, and patching techniques to enhance temporal sequence representation. Notably, the contributions of [263] [242] [165] have collectively advanced the development of generalized strategies for spatio-temporal problem-solving through transformer-based architectures. Alternative methodologies for handling higher-order representations of multivariate time series have been proposed in [170]. Here, the self-attention mechanism is adapted to accommodate three-axis tensor representations by employing a Low-Rank Approximation with Kronecker Decomposition. The capture of dependencies in sequential data could be improved by combining these techniques with the proposed ASM-based method.

Nevertheless, a close conceptual link between these methods and core NLP principles is demonstrated in this thesis. Within the proposed framework, a general approach is now outlined that could potentially be transferred to other multivariate time series processing tasks. It should be emphasized that approaches that did not work in the examples (e.g. TM in ASMs) may now do so in domains such as weather forecasting or energy consumption prediction. For the following a multivariate time series is defined as $X \in \mathbb{R}^{|C| \times \mathbb{T} \times \mathbb{F}}$ with the multivariate variables $c_i \in C$, the time steps t and the features $\mathbb{F}$. For example, c_i could now correspond to different weather stations, $\mathbb{F}$ various measured physical quantities and $\mathbb{T}$ the measured time period.

Meteorological forecasting is used as the example application. For stacked features and variables, processing by recurrent or pretrained transformers is straightforward. These architectures can take the input data directly, allowing training through established methods such as masking, trend prediction, and trend alignment—techniques that have proven robust across domains, though with limited flexibility. A key advantage of the ASM architecture lies in its inherent extensibility. New meteorological stations can be integrated without full retraining or discarding previously learned spatial information. A critical aspect in this context is the determination of optimal values for Δt, which are expected to exhibit significant variability across different domains.

Nevertheless, despite the increased complexity, their application remains advantageous due to the numerous benefits delineated in Chapter 8.

For the effective utilization of ASM architectures, it is advisable to employ the C-CBOS and X-CBOS algorithms to construct domain-specific, contextualized vector representations of the relevant variables—such as meteorological observation stations—within the embedding matrix E. This approach is especially relevant given the inherent complexity of spatial relationships among variables, as highlighted by [245]. Two primary methodologies can be employed in this context. First, a sliding window approach can be utilized to process variable sequences, incorporating one or more features to facilitate regression-based forecasting or movement prediction. Alternatively, multiple variables can be integrated to classify or estimate one or more attributes of a target variable. For instance, a classification model may utilize historical and future temperature observations spanning a one-week period to estimate the temperature on a specific day. Similarly, meteorological data from multiple weather stations on a given day can be aggregated to infer the temperature at a station that is masked.

The generated embeddings serve as a foundational representation for input into an ASM architecture. To establish a coherent temporal context, a sliding window is applied to the time series. The spatial relationships among $c_i \in C$ are encoded by concatenating the corresponding (contextualized) embeddings at each time step within the sliding window. To preserve temporal ordering, positional encodings are incorporated, facilitating the model's ability to distinguish between different time steps. For instance, at a given time step, meteorological data from all available weather stations are concatenated in a structured manner and arranged according to the defined sliding window size.

Analogous to positional embeddings, temporal dependencies between features are encoded through learned shifting vectors (i.e. different interval granularities). Furthermore, the measurement data of each weather station are embedded as learned feature vector representations (i.e. using $F^{\langle \text{FEW} \rangle}$), functioning as shifts on the respective station-specific embedding vectors. Indicator correlations are also covered here. These structured sliding window representations can then be processed by the ASM in two primary ways: (i) for predictive tasks, or (ii) within a decoder-based architecture to generate future states.

A generalized framework is established by letting the ASM process (pretrained) contextualized embeddings for spatial dependencies, representing temporal information with learned shifts, and distinguishing time steps via positional embeddings and flattening. The generalized approach to using time series in ASM-based foundation models is illustrated in Figure 9.2. One possibility not illustrated here is the integration of Doc2Vec-adapted embeddings (by reconstructing inputs from dense vector representations) in order to provide macro-level information—such as climate context in the case of weather data.

In this thesis, the applicability of this method in the domain of financial time series analysis, specifically in modeling stock price data, has been demonstrated. By leveraging the ASM framework, it has been shown that pretrained contextualized embeddings can enhance the capture of intricate interdependencies between different assets, market conditions, and temporal trends.

Given the versatility of ASMs in processing multivariate time series data, it is anticipated that this approach can be successfully extended to a variety of other domains, including meteorology, healthcare, and engineering, where sequential data plays a crucial role in decision-making and forecasting. Further adaptations from the NLP area for time series processing, especially with regard to foundation models, are hoped for. It is also hoped that the findings will encourage further research and experimentation in this direction, leading to advancements in time-series modeling across diverse application areas.

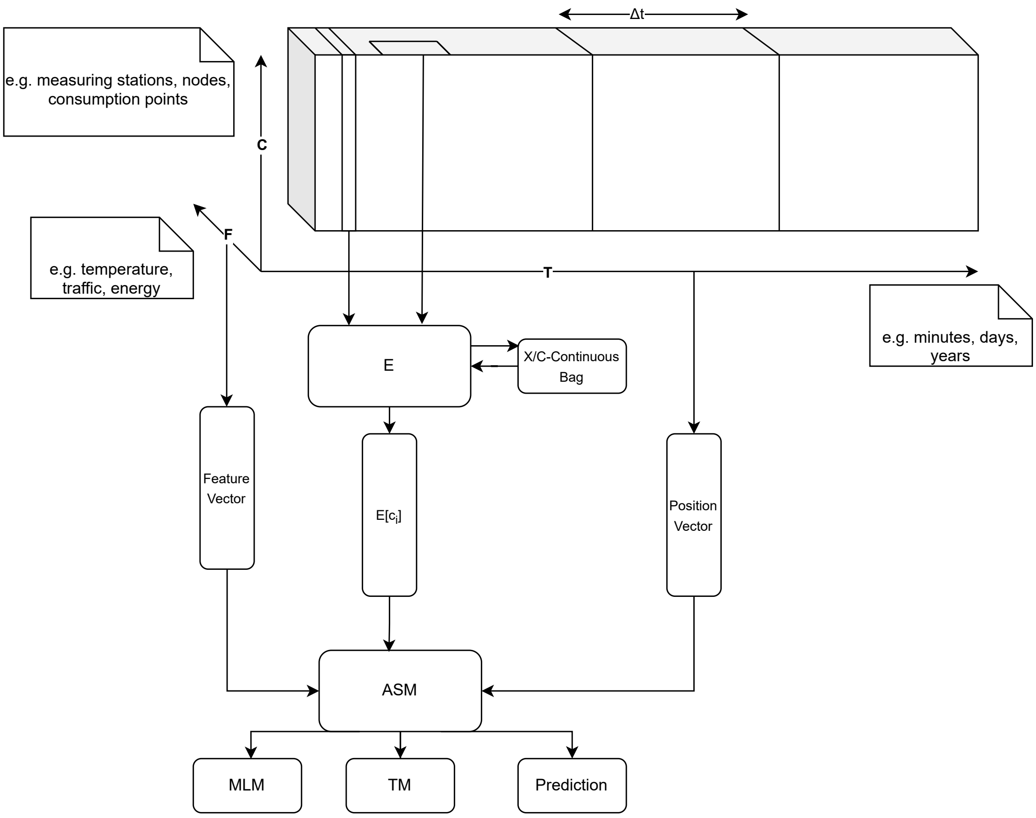

Δt
e.g. measuring stations, nodes, consumption points
C
F
e.g. temperature, traffic, energy
T
e.g. minutes, days, years
E
X/C-Continuous Bag
Feature Vector
E[c_i]
Position Vector
ASM
MLM
TM
Prediction

FIGURE 9.2: Sketch of the general approach of using ASMs as foundation models for time series data.

9.3 Summary of the Main Results and Contributions

In this dissertation, a comprehensive experimental strategy was carried out to transfer key concepts from NLP to the field of quantitative financial time series analysis. The study was motivated by a significant gap in the existing literature regarding the use of NLP strategies, models, and findings in other related domains. This gap was primarily reflected in the lack of generalizable, extensible, and pretrained models capable of capturing both spatial relationships and temporal dependencies in large-scale financial datasets. To the best of the authors knowledge, this thesis is the first to address this issue by proposing a class of models referred to as ASMs.

These models were inspired by the transformer-based architecture of LLMs and were designed to enable the adaptation to various markets, temporal resolutions, and application domains. Furthermore, this work represents the first application of SMC and SPE, as well as the unsuccessful TM task, as pretraining tasks for time series—especially in the concrete form later realized in the ASMs. To evaluate the suitability of the pretraining+finetuning paradigm, spatial, temporal, and spatio-temporal masking strategies were tested in various variations within the transformer-based $F^{\langle \mathrm{T} \rangle}$ approaches. With regard to representation learning, the research gap concerning unsupervised SMC- and SME-based methods for contextualized embeddings was addressed in the proposed S2V models. These methods were tested on both the spatial and temporal axes to generate stock embeddings. In parallel, document-level embedding methods such as Doc2Vec—referred to in this work as QMSEs—were adapted. This thesis is the first to introduce contextualized vectors for summarizing macroeconomic situations and evaluating them distinctly, rather than as part of the model pipeline. These approaches were not merely used for prediction but aimed to learn comprehensive representations of market structures. The goal of the foundation ASM models design was to leverage pretraining across a wide range of datasets, allowing the models to internalize inter-stock dependencies and indicator-driven patterns before being fine-tuned on specific forecasting targets such as SMP or SPP. Further novelties of this thesis include the use of CWRNNs for quantitative stock data, results for V+L adapted models, and the application of adapted LLMs on token level or as transformer backbones where stock data were input as embeddings.

The experimental results and the subsequent analytical discussions yielded several key insights. First, an embedding vector-based spatio-temporal representation in the proposed Stock2Sentence approach, combined with modular integration of indicators, temporal windows, and optional embeddings such as S2V and Doc2Vec, was shown to provide a highly flexible and extensible modeling framework. Second, the adaptation of masking-based tasks from NLP proved particularly effective for pretraining on financial time series, whereas direct analogs of NSP—i.e. trend matching—failed to be adapted. Third, the utility of pretraining was supported by improvements in classification performance and reductions in overfitting, especially when diverse datasets were used in the pretraining phase. An important insight was that although pretraining encodes parts of the downstream optimization process (i.e. learning of indicator and relationship information) into the model, it may not lead to performance improvements due to limited optimization margins typical in SF models. Fourth, the use of context embeddings via Doc2Vec (QMSE) enabled implicit encoding of macroeconomic conditions, offering potential advantages for downstream tasks such as risk management and portfolio allocation. Applications in these areas are suggested by the near-perfect performance of ASMs as SMC models. It was shown that these embeddings can serve as training regularizers and, although not always beneficial, can be used particularly in unstable models. Finally, the combination of ASMs, QMSEs, and S2V representations was used to present a method for creating foundation models for any multivariate time series that are extendable, generalizable, and pretrainable, and may prove useful in other domains.

An empirical validation was carried out across multiple temporal granularities and datasets (All$^{(2010:)}$, S&P-500 🇺🇸), illustrating both strengths and limitations of the proposed techniques. In conclusion, the foundational paradigm of pretraining followed by finetuning, which has become central in NLP, was successfully adapted and validated for application in quantitative finance. A modeling framework is established and its potential across scenarios is demonstrated, laying groundwork for LLM/NLP-inspired time-series models that incorporate richer modalities and aim to improve generalization in dynamic financial settings. All adapted NLP concepts are listed in Table 9.1 for comparison.

Table 9.1: Conceptual comparison between NLP and stock forecasting methods

NLP Concept	**Adaptation for Stock Forecasting**
Pretraining text corpus	All$^{(2010:)}$
(Small) Finetuning dataset	(interday) S&P-500 🇺🇸
Word token embedding	Market snapshot $\bar{X}$
(W2V) Word embedding matrix	$F^{\langle \mathrm{LL} \rangle}$
(W2V) Word embedding matrix	S2V embeddings
W2V Vocabulary	SMC input labels in S2V
LLM context-sensitive embeddings	ASM embeddings
CBOW / SG	CBOS / Stock-SG
LLM (foundation model)	ASM (foundation model)
MLM	Masking tasks
NSP	TM
Positional encoding	Time-step encoding
Next-token prediction (causal LM)	SMP / SPP
CLM	Π integration (QMSE / T data / fundamental data)
Finetuning / downstream task[a]	SMP / SPP
Zero-shot / few-shot learning	Market shift handling
Transformer Encoder/Decoder	Transformer for multivariate time series forecasting
Recurrent Transformer	Block-recurrent Transformer for large Δt
(Implicit) Hierarchical models	Multi-frequency (multi-timeframe) representation
Documents	Market Situations
Doc2Vec (document embedding)	QMSEs
Attention mechanism	Inter-stock correlation weighting
Global attention	Market-wide context modeling

[a] e.g., summary generation, sentiment classification

[1] Klaus Adam, Albert Marcet and Juan Pablo Nicolini. 'Stock Market Volatility and Learning'. In: *The Journal of Finance* 71.1 (2016), pp. 33–82. URL: http://www.jstor.org/stable/43869095.

[2] Ryo Akita, Akira Yoshihara, Takashi Matsubara and Kuniaki Uehara. 'Deep learning for stock prediction using numerical and textual information'. In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science*. 2016, pp. 1–6. DOI: 10.1109/ICIS.2016.7550882.

[3] Usman Ali and David Hirshleifer. 'Shared analyst coverage: Unifying momentum spillover effects'. In: *Journal of Financial Economics* 136.3 (2020), pp. 649–675. ISSN: 0304-405X. DOI: https://doi.org/10.1016/j.jfineco.2019.10.007. URL: https://www.sciencedirect.com/science/article/pii/S0304405X19302533.

[4] Gary Ang and Ee-Peng Lim. 'Guided Attention Multimodal Multitask Financial Forecasting with Inter-Company Relationships and Global and Local News'. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*. Ed. by Smaranda Muresan, Preslav Nakov and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 6313–6326. DOI: 10.18653/v1/2022.acl-long.437. URL: https://aclanthology.org/2022.acl-long.437.

[5] Wei Bao, Jun Yue and Yulei Rao. 'A deep learning framework for financial time series using stacked autoencoders and long-short term memory'. In: *PLOS ONE* 12.7 (July 2017), pp. 1–24. DOI: 10.1371/journal.pone.0180944. URL: https://doi.org/10.1371/journal.pone.0180944.

[6] Marco Baroni, Georgiana Dinu and Germán Kruszewski. 'Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors'. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 238–

247. DOI: 10.3115/v1/P14-1023. URL: https://aclanthology.org/P14-1023.

[7] Iz Beltagy, Matthew E. Peters and Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. arXiv: 2004.05150 [cs.CL]. URL: https://arxiv.org/abs/2004.05150.

[8] Amanda Bertsch, Uri Alon, Graham Neubig and Matthew R. Gormley. 'Unlimiformer: long-range transformers with unlimited length input'. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc., 2023.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever and Dario Amodei. 'Language Models are Few-Shot Learners'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[10] Aydar Bulatov, Yuri Kuratov, Yermek Kapushev and Mikhail Burtsev. 'Beyond Attention: Breaking the Limits of Transformer Context Length with Recurrent Memory'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.16 (Mar. 2024), pp. 17700–17708. DOI: 10.1609/aaai.v38i16.29722. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29722.

[11] José Camacho-Collados and Roberto Navigli. 'Find the word that does not belong: A Framework for an Intrinsic Evaluation of Word Vector Representations'. In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational

Linguistics, Aug. 2016, pp. 43–50. DOI: 10.18653/v1/W16-2508. URL: https://aclanthology.org/W16-2508.

[12] Defu Cao, Yixiang Zheng, Parisa Hassanzadeh, Simran Lamba, Xiaomo Liu and Yan Liu. 'Large Scale Financial Time Series Forecasting with Multi-faceted Model'. In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pp. 472–480. ISBN: 9798400702402. DOI: 10.1145/3604237.3626868. URL: https://doi.org/10.1145/3604237.3626868.

[13] Lele Cao, Vilhelm von Ehrenheim, Mark Granroth-Wilding, Richard Anselmo Stahl, Andrew McCornack, Armin Catovic and Dhiana Deva Cavacanti Rocha. 'CompanyKG: A Large-Scale Heterogeneous Graph for Company Similarity Quantification'. In: *IEEE Transactions on Big Data* (2024), pp. 1–12. ISSN: 2372-2096. DOI: 10.1109/tbdata.2024.3407573. URL: http://dx.doi.org/10.1109/TBDATA.2024.3407573.

[14] Deeksha Chandola, Akshit Mehta, Shikha Singh, Vinay Anand Tikkiwal and Himanshu Agrawal. 'Forecasting Directional Movement of Stock Prices using Deep Learning'. In: *Annals of Data Science* 10.5 (Oct. 2023), pp. 1361–1378. ISSN: 2198-5812. DOI: 10.1007/s40745-022-00432-6. URL: https://doi.org/10.1007/s40745-022-00432-6.

[15] Ching Chang, Wei-Yao Wang, Wen-Chih Peng, Tien-Fu Chen and Sagar Samtani. 'Align and Fine-Tune: Enhancing LLMs for Time-Series Forecasting'. In: *Workshop on Time Series in the Age of Large Models at the Conference on Neural Information Processing Systems, 2024*. 2024. URL: https://openreview.net/forum?id=AaRCmJieG4.

[16] Kinjal Chaudhari and Ankit Thakkar. 'Data fusion with factored quantization for stock trend prediction using neural networks'. In: *Information Processing and Management* 60.3 (2023), p. 103293. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2023.103293. URL: https://www.sciencedirect.com/science/article/pii/S0306457323000304.

[17] Hila Chefer, Shir Gur and Lior Wolf. 'Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers'. In: *2021 IEEE/CVF International Conference on Computer Vision*. 2021, pp. 387–396. DOI: 10.1109/ICCV48922.2021.00045.

[18] Deli Chen, Yanyan Zou, Keiko Harimoto, Ruihan Bao, Xuancheng Ren and Xu Sun. 'Incorporating Fine-grained Events in Stock Movement Prediction'. In: *Proceedings of the Second Workshop on Economics and Natural Language Processing*. Ed. by Udo Hahn, Véronique Hoste and Zhu Zhang. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 31–40. DOI: 10.18653/v1/D19-5105. URL: https://aclanthology.org/D19-5105/.

[19] Jia Chen, Tao Chen, Mengqi Shen, Yunhai Shi, Dongjing Wang and Xin Zhang. 'Gated three-tower transformer for text-driven stock market prediction'. In: *Multimedia Tools and Applications* 81.21 (Sept. 2022), pp. 30093–30119. ISSN: 1573-7721. DOI: 10.1007/s11042-022-11908-1. URL: https://doi.org/10.1007/s11042-022-11908-1.

[20] Jiahao Chen, Liang Xie, Wenjing Lin, Yuchen Wu and Haijiao Xu. 'Multi-Granularity Spatio-temporal Correlation Networks for Stock Trend Prediction'. In: *IEEE Access* (2024), pp. 1–1. DOI: 10.1109/ACCESS.2024.3393774.

[21] Kai Chen, Yi Zhou and Fangyan Dai. 'A LSTM-based method for stock returns prediction: A case study of China stock market'. In: *2015 IEEE International Conference on Big Data* (2015), pp. 2823–2824.

[22] Yakun Chen, Xianzhi Wang and Guandong Xu. *GATGPT: A Pre-trained Large Language Model with Graph Attention Network for Spatiotemporal Imputation*. 2023. arXiv: 2311.14332 [cs.LG]. URL: https://arxiv.org/abs/2311.14332.

[23] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng and Jingjing Liu. 'UNITER: UNiversal Image-TExt Representation Learning'. In: *Computer Vision – European Conference on*

Computer Vision 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 104–120. ISBN: 978-3-030-58576-1. DOI: 10.1007/978-3-030-58577-8_7. URL: https://doi.org/10.1007/978-3-030-58577-8_7.

[24] Yingmei Chen, Zhongyu Wei and Xuanjing Huang. 'Incorporating Corporation Relationship via Graph Convolutional Neural Networks for Stock Price Prediction'. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Torino, Italy: Association for Computing Machinery, 2018, pp. 1655–1658. ISBN: 9781450360142. DOI: 10.1145/3269206.3269269. URL: https://doi.org/10.1145/3269206.3269269.

[25] You-Sin Chen, Chu-Lan Michael Kao, Po-Hsien Liu and Vincent S. Tseng. 'Extracting Stock Predictive Information in Mutual Fund Managers' Portfolio Decisions Through Machine Learning with Hypergraph'. In: *Computational Economics* (July 2024). ISSN: 1572-9974. DOI: 10.1007/s10614-024-10673-7. URL: https://doi.org/10.1007/s10614-024-10673-7.

[26] Rui Cheng and Qing Li. 'Modeling the Momentum Spillover Effect for Stock Prediction via Attribute-Driven Graph Attention Networks'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.1 (May 2021), pp. 55–62. DOI: 10.1609/aaai.v35i1.16077. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16077.

[27] Yang Qing Wang Chenwei. 'A Study on Forecast of Global Stock Indices Based on Deep LSTM Neural Network'. In: *Statistical Research* 36.3, 65 (2019), p. 65. DOI: 10.19343/j.cnki.11-1302/c.2019.03.006. URL: https://tjyj.stats.gov.cn/EN/abstract/article_5196.shtml.

[28] Donghee Choi, Jinkyu Kim, Mogan Gim, Jinho Lee and Jaewoo Kang. 'DeepClair: Utilizing Market Forecasts for Effective Portfolio Selection'. In: *Proceedings of the 33rd ACM International Conference on Information and*

Knowledge Management. Boise, ID, USA: ACM, 2024, October 21–25. ISBN: 979-8-4007-0436-9. DOI: 10.1145/3627673.3680008.

[29] Jooweon Choi, Shiyong Yoo, Xiao Zhou and Youngbin Kim. 'Hybrid Information Mixing Module for Stock Movement Prediction'. In: *IEEE Access* 11 (2023), pp. 28781–28790. DOI: 10.1109/ACCESS.2023.3258695.

[30] Francesco Colasanto, Luca Grilli, Domenico Santoro and Giovanni Villani. 'AlBERTino for stock price prediction: a Gibbs sampling approach'. In: *Information Sciences* 597 (2022), pp. 341–357. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.03.051. URL: https://www.sciencedirect.com/science/article/pii/S002002552200264X.

[31] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le and Ruslan Salakhutdinov. 'Transformer-XL: Attentive Language Models beyond a Fixed-Length Context'. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 2978–2988. DOI: 10.18653/V1/P19-1285. URL: https://doi.org/10.18653/v1/p19-1285.

[32] Divyanshu Daiya and Che Lin. 'Stock Movement Prediction and Portfolio Management via Multimodal Learning with Transformer'. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2021, pp. 3305–3309. DOI: 10.1109/ICASSP39728.2021.9414893.

[33] Divyanshu Daiya, Monika Yadav and Harshit Singh Rao. 'Diffstock: Probabilistic Relational Stock Market Predictions Using Diffusion Models'. In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2024, pp. 7335–7339. DOI: 10.1109/ICASSP48485.2024.10446690.

[34] Hong N. Dao, Wang ChuanYuan, Aoshi Suzuki, Hitomi Sudo, Li Ye and Debopriyo Roy. 'AI in Stock Market Forecasting: A Bibliometric Analysis'. In: *SHS Web of Conferences* 194 (2024). The 6th International Conference on ICT Integration in Technical Education, Section: Intelligent Applications in Society., p. 01003. DOI: 10.1051/shsconf/202419401003. URL: https://doi.org/10.1051/shsconf/202419401003.

[35] Abhimanyu Das, Weihao Kong, Rajat Sen and Yichen Zhou. 'A decoder-only foundation model for time-series forecasting'. In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 10148–10167. URL: https://proceedings.mlr.press/v235/das24c.html.

[36] Richard A. DeFusco, Dennis W. McLeavey, Jerald E. Pinto and David E. Runkle. *Quantitative Investment Analysis*. 3rd ed. Wiley, 2015. ISBN: 9781119101020.

[37] Azad Deihim, Eduardo Alonso and Dimitra Apostolopoulou. 'STTRE: A Spatio-Temporal Transformer with Relative Embeddings for multivariate time series forecasting'. In: *Neural Networks* 168 (2023), pp. 549–559. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2023.09.039. URL: https://www.sciencedirect.com/science/article/pii/S0893608023005361.

[38] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan and Huajun Chen. 'Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network'. In: *Companion Proceedings of The 2019 World Wide Web Conference*. San Francisco, USA: Association for Computing Machinery, 2019, pp. 678–685. ISBN: 9781450366755. DOI: 10.1145/3308560.3317701. URL: https://doi.org/10.1145/3308560.3317701.

[39] Daniel Deutsch and Dan Roth. 'Understanding the Extent to which Content Quality Metrics Measure the Information Quality of Summaries'. In: *Proceedings of the 25th Conference on Computational Natural Language Learning*. Ed. by Arianna Bisazza and Omri Abend. Online: Association for Computational Linguistics, Nov. 2021, pp. 300–309. DOI: 10.18653/v1/2021.conll-1.24. URL: https://aclanthology.org/2021.conll-1.24/.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Jill Burstein, Christy Doran and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423/.

[41] Luca Di Persio and Oleksandr Honchar. 'Artificial neural networks architectures for stock price prediction: Comparisons and applications'. In: *International Journal of Circuits, Systems and Signal Processing* 10.2016 (2016), pp. 403–413.

[42] Guangyu Ding and Liangxi Qin. 'Study on the prediction of stock price based on the associated network model of LSTM'. In: *International Journal of Machine Learning and Cybernetics* 11 (June 2020). DOI: 10.1007/s13042-019-01041-1.

[43] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo and Jian Guo. 'Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction'. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. Ed. by Christian Bessiere. Special Track on AI in FinTech. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 4640–4646. DOI: 10.24963/ijcai.2020/640. URL: https://doi.org/10.24963/ijcai.2020/640.

[44] Xiao Ding, Yue Zhang, Ting Liu and Junwen Duan. 'Deep Learning for Event-Driven Stock Prediction'. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, 2015, pp. 2327–2333. ISBN: 9781577357384.

[45] Yujie Ding, Shuai Jia, Tianyi Ma, Bingcheng Mao, Xiuze Zhou, Liuliu Li and Dongming Han. *Integrating Stock Features and Global Information via Large Language Models for Enhanced Stock Return Prediction*. 2023. arXiv: 2310.05627 [cs.CL]. URL: https://arxiv.org/abs/2310.05627.

[46] Joy Dip Das, Ruppa K. Thulasiram, Christopher Henry and Aerambamoorthy Thavaneswaran. 'Encoder–Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction'. In: *Journal of Risk and Financial Management* 17.5 (2024). ISSN: 1911-8074. DOI: 10.3390/jrfm17050200. URL: https://www.mdpi.com/1911-8074/17/5/200.

[47] Rian Dolphin, Barry Smyth and Ruihai Dong. 'A Machine Learning Approach to Industry Classification in Financial Markets'. In: *Artificial Intelligence and Cognitive Science*. Ed. by Luca Longo and Ruairi O'Reilly. Cham: Springer Nature Switzerland, 2023, pp. 81–94. ISBN: 978-3-031-26438-2.

[48] Rian Dolphin, Barry Smyth and Ruihai Dong. 'Contrastive Learning of Asset Embeddings from Financial Time Series'. In: *Proceedings of the 5th ACM International Conference on AI in Finance*. Brooklyn, NY, USA: Association for Computing Machinery, 2024, pp. 379–387. ISBN: 9798400710810. DOI: 10.1145/3677052.3698610. URL: https://doi.org/10.1145/3677052.3698610.

[49] Rian Dolphin, Barry Smyth and Ruihai Dong. *Industry Classification Using a Novel Financial Time-Series Case Representation*. 2023. arXiv: 2305.00245 [cs.LG].

[50] Rian Dolphin, Barry Smyth and Ruihai Dong. ‘Stock Embeddings: Representation Learning for Financial Time Series’. In: *Engineering Proceedings* 39.1 (2023). ISSN: 2673-4591. DOI: 10.3390/engproc2023039030. URL: https://www.mdpi.com/2673-4591/39/1/30.

[51] Yingzhe Dong, Da Yan, Abdullateef Ibrahim Almudaifer, Sibo Yan, Zhe Jiang and Yang Zhou. ‘BELT: A Pipeline for Stock Price Prediction Using News’. In: *2020 IEEE International Conference on Big Data*. 2020, pp. 1137–1146. DOI: 10.1109/BigData50022.2020.9378345.

[52] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby. ‘An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale’. In: *9th International Conference on Learning Representations, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[53] Kelvin Du, Rui Mao, Frank Xing and Erik Cambria. ‘A Dynamic Dual-Graph Neural Network for Stock Price Movement Prediction’. In: *2024 International Joint Conference on Neural Networks*. 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650440.

[54] Xin Du and Kumiko Tanaka-Ishii. ‘Stock Embeddings Acquired from News Articles and Price History, and an Application to Portfolio Optimization’. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 3353–3363. DOI: 10.18653/v1/2020.acl-main.307. URL: https://aclanthology.org/2020.acl-main.307.

[55] Junwen Duan, Yue Zhang, Xiao Ding, Ching-Yun Chang and Ting Liu. ‘Learning Target-Specific Representations of Financial News Documents For Cumulative Abnormal Return Prediction’. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Emily M. Bender, Leon Derczynski and Pierre Isabelle. Santa Fe, New Mexico, USA:

Association for Computational Linguistics, Aug. 2018, pp. 2823–2833. URL: https://aclanthology.org/C18-1239.

[56] Jithin Eapen, Doina Bein and Abhishek Verma. 'Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction'. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference*. 2019, pp. 0264–0270. DOI: 10.1109/CCWC.2019.8666592.

[57] Jithin Eapen, Doina Bein and Abhishek Verma. 'Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction'. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 2019, pp. 0264–0270. DOI: 10.1109/CCWC.2019.8666592.

[58] J. L. Elman. 'Finding Structure In Time'. In: *Cognitive Science* 14 (1990), pp. 179–211.

[59] Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. 'A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise'. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[60] Eugene F. Fama. 'Efficient Capital Markets: A Review of Theory and Empirical Work'. In: *The Journal of Finance* 25.2 (1970), pp. 383–417. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325486 (visited on 06/02/2023).

[61] Chenyou Fan, Tianqi Pang and Aimin Huang. 'Pre-trained Financial Model for Price Movement Forecasting'. In: *Neural Information Processing*. Ed. by Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li and Chaojie Li. Singapore: Springer Nature Singapore, 2024, pp. 216–229. ISBN: 978-981-99-8184-7.

[62] Jinyong Fan and Yanyan Shen. 'StockMixer: A Simple Yet Strong MLP-Based Architecture for Stock Price Forecasting'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.8 (Mar. 2024), pp. 8389–8397. DOI: 10.1609/aaai.v38i8.28681. URL: https://ojs.aaai.org/index.php/AAAI/article/view/28681.

[63] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi and Chris Dyer. 'Problems With Evaluation of Word Embeddings Using Word Similarity Tasks'. In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 30–35. DOI: 10.18653/v1/W16-2506. URL: https://aclanthology.org/W16-2506/.

[64] Kamaladdin Fataliyev and Wei Liu. 'MCASP: Multi-Modal Cross Attention Network for Stock Market Prediction'. In: *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*. Ed. by Smaranda Muresan, Vivian Chen, Kennington Casey, Vandyke David, Dethlefs Nina, Inoue Koji, Ekstedt Erik and Ultes Stefan. Melbourne, Australia: Association for Computational Linguistics, Nov. 2023, pp. 67–77. URL: https://aclanthology.org/2023.alta-1.7.

[65] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun and Tat-Seng Chua. 'Enhancing Stock Movement Prediction with Adversarial Training'. In: *International Joint Conference on Artificial Intelligence*. 2018. URL: https://api.semanticscholar.org/CorpusID:173991126.

[66] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu and Tat-Seng Chua. 'Temporal Relational Ranking for Stock Prediction'. In: *ACM Transactions on Information Systems* 37.2 (Mar. 2019). ISSN: 1046-8188. DOI: 10.1145/3309547. URL: https://doi.org/10.1145/3309547.

[67] Xinghong Fu, Masanori Hirano and Kentaro Imajo. *Financial Fine-tuning a Large Time Series Model*. 2024. arXiv: 2412.09880 [q-fin.CP]. URL: https://arxiv.org/abs/2412.09880.

[68] Xavier Gabaix, Ralph Koijen and Motohiro Yogo. 'Asset Embeddings'. In: *SSRN Electronic Journal* (Jan. 2023). DOI: 10.2139/ssrn.4507511.

[69] Siyu Gao, Yunbo Wang and Xiaokang Yang. 'StockFormer: Learning Hybrid Trading Machines with Predictive Coding'. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. Ed. by Edith Elkind. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 4766–4774. DOI: 10.24963/ijcai.2023/530. URL: https://doi.org/10.24963/ijcai.2023/530.

[70] Tingwei Gao, Yueting Chai and Yi Liu. 'Applying long short term momory neural networks for predicting stock closing price'. In: *2017 8th IEEE International Conference on Software Engineering and Service Science*. 2017, pp. 575–578. DOI: 10.1109/ICSESS.2017.8342981.

[71] Yuan Gao, Haokun Chen, Xiang Wang, Zhicai Wang, Xue Wang, Jinyang Gao and Bolin Ding. 'DiffsFormer: A Diffusion Transformer on Stock Factor Augmentation'. In: *CoRR* abs/2402.06656 (2024). DOI: 10.48550/ARXIV.2402.06656. arXiv: 2402.06656. URL: https://doi.org/10.48550/arXiv.2402.06656.

[72] Azul Garza, Cristian Challu and Max Mergenthaler-Canseco. *TimeGPT-1*. 2024. arXiv: 2310.03589 [cs.LG]. URL: https://arxiv.org/abs/2310.03589.

[73] Menelik Geremew and Francois Gourio. 'Seasonal and Business Cycles of U.S. Employment'. In: *Economic Perspectives* 3 (2018), pp. 1–28. URL: https://EconPapers.repec.org/RePEc:fip:fedhep:00032.

[74] Abdul Haluk Batur Gezici and Emre Sefer. 'Deep Transformer-Based Asset Price and Direction Prediction'. In: *IEEE Access* 12 (2024), pp. 24164–24178. DOI: 10.1109/ACCESS.2024.3358452.

[75] Marjan Ghazvininejad, Omer Levy, Yinhan Liu and Luke Zettlemoyer. 'Mask-Predict: Parallel Decoding of Conditional Masked Language Models'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019. DOI: 10.18653/v1/D19-1633. URL: https://aclanthology.org/D19-1633.

[76] Gyözö Gidófalvi. 'Using News Articles to Predict Stock Price Movements'. In: 2001. URL: https://api.semanticscholar.org/CorpusID:17308076.

[77] Xavier Glorot and Yoshua Bengio. 'Understanding the difficulty of training deep feedforward neural networks'. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[78] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2024. arXiv: 2312.00752 [cs.LG]. URL: https://arxiv.org/abs/2312.00752.

[79] Jingyi Gu, Junyi Ye, Ajim Uddin and Guiling Wang. 'DySTAGE: Dynamic Graph Representation Learning for Asset Pricing via Spatio-Temporal Attention and Graph Encodings'. In: *Proceedings of the 5th ACM International Conference on AI in Finance*. Brooklyn, NY, USA: Association for Computing Machinery, 2024, pp. 388–396. ISBN: 9798400710810. DOI: 10.1145/3677052.3698680. URL: https://doi.org/10.1145/3677052.3698680.

[80] Jingyi Gu, Junyi Ye, Guiling Wang and Wenpeng Yin. 'Adaptive and Explainable Margin Trading via Large Language Models on Portfolio Management'. In: *Proceedings of the 5th ACM International Conference on AI in*

Finance. Brooklyn, NY, USA: Association for Computing Machinery, 2024, pp. 248–256. ISBN: 9798400710810. DOI: 10.1145/3677052.3698681. URL: https://doi.org/10.1145/3677052.3698681.

[81] Shihao Gu, Bryan T. Kelly and Dacheng Xiu. *Autoencoder Asset Pricing Models*. Tech. rep. Available at SSRN: https://ssrn.com/abstract=3335536 or http://dx.doi.org/10.2139/ssrn.3335536. Yale ICF Working Paper No. 2019-04, Chicago Booth Research Paper No. 19-24, Sept. 2019.

[82] Jian Guo and Heung-Yeung Shum. *Large Investment Model*. 2024. arXiv: 2408.10255 [q-fin.ST]. URL: https://arxiv.org/abs/2408.10255.

[83] Kaiming He, Ross Girshick and Piotr Dollar. 'Rethinking ImageNet Pre-Training'. In: *2019 IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4917–4926. DOI: 10.1109/ICCV.2019.00502.

[84] Liwen He and Wentao Xu. 'Confrontation-LSTM Network for stock trend prediction'. In: *2023 China Automation Congress*. 2023, pp. 3017–3021. DOI: 10.1109/CAC59555.2023.10451268.

[85] Qi-Qiao He, Shirley Weng In Siu and Yain-Whar Si. 'Instance-based deep transfer learning with attention for stock movement prediction'. In: *Applied Intelligence* 53.6 (Mar. 2023), pp. 6887–6908. ISSN: 1573-7497. DOI: 10.1007/s10489-022-03755-2. URL: https://doi.org/10.1007/s10489-022-03755-2.

[86] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[87] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* Vol. 9 (Dec. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[88] Ehsan Hoseinzade, Saman Haratizadeh and Arash Khoeini. *U-CNNpred: A Universal CNN-based Predictor for Stock Markets*. 2019. arXiv: 1911.12540 [cs.LG].

[89] Jeremy Howard and Sebastian Ruder. 'Universal Language Model Fine-tuning for Text Classification'. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '18)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://aclanthology.org/P18-1031/.

[90] Ting-Wei Hsu, Chung-Chi Chen, Hen-Hsen Huang and Hsin-Hsi Chen. 'Semantics-Preserved Data Augmentation for Aspect-Based Sentiment Analysis'. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4417–4422. DOI: 10.18653/v1/2021.emnlp-main.362.

[91] Xiaokang Hu. 'Stock Price Prediction Based on Temporal Fusion Transformer'. In: *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence*. 2021, pp. 60–66. DOI: 10.1109/MLBDBI54094.2021.00019.

[92] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu and Tie-Yan Liu. 'Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction'. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 261–269. ISBN: 9781450355810. DOI: 10.1145/3159652.3159690. URL: https://doi.org/10.1145/3159652.3159690.

[93] Jieyun Huang, Yunjia Zhang, Jialai Zhang and Xi Zhang. 'A Tensor-Based Sub-Mode Coordinate Algorithm for Stock Prediction'. In: *2018*

IEEE Third International Conference on Data Science in Cyberspace. 2018, pp. 716–721. DOI: 10.1109/DSC.2018.00114.

[94] DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer and Behnam Neyshabur. 'Block-Recurrent Transformers'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 33248–33261. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/d6e0bbb9fc3f4c10950052ec2359355c-Paper-Conference.pdf.

[95] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen and Karl Aberer. *Efficient Integration of Multi-Order Dynamics and Internal Dynamics in Stock Movement Prediction*. 2022. arXiv: 2211.07400 [q-fin.ST]. URL: https://arxiv.org/abs/2211.07400.

[96] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen and Karl Aberer. 'Efficient Integration of Multi-Order Dynamics and Internal Dynamics in Stock Movement Prediction'. In: Proceedings of the 16th ACM International Conference on Web Search and Data Mining (2023). Publisher Copyright: © 2023 ACM.; 16th ACM International Conference on Web Search and Data Mining, WSDM 2023 ; Conference date: 27-02-2023 Through 03-03-2023. Feb. 2023, pp. 850–858. DOI: 10.1145/3539597.3570427.

[97] Yoontae Hwang, Junhyeong Lee, Daham Kim, Seunghwan Noh, Joohwan Hong and Yongjae Lee. 'SimStock: Representation Model for Stock Similarities'. In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pp. 533–540. ISBN: 9798400702402. DOI: 10.1145/3604237.3626888. URL: https://doi.org/10.1145/3604237.3626888.

[98] Shibal Ibrahim, Max Tell and Rahul Mazumder. 'Dyn-GWN: Time-Series Forecasting using Time-varying Graphs with Applications to Finance and Traffic Prediction'. In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pp. 167–175. ISBN: 9798400702402. DOI: 10.1145/3604237.3626864. URL: https://doi.org/10.1145/3604237.3626864.

[99] Bruce I. Jacobs and Kenneth N. Levy. 'Calendar Anomalies: Abnormal Returns at Calendar Turning Points'. In: *Financial Analysts Journal* 44.6 (1988), pp. 28–39. DOI: 10.2469/faj.v44.n6.28.

[100] Jihyeong Jeon, Jiwon Park, Chanhee Park and U Kang. 'FreQuant: A Reinforcement-Learning based Adaptive Portfolio Optimization with Multi-frequency Decomposition'. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 1211–1221. ISBN: 9798400704901. DOI: 10.1145/3637528.3671668. URL: https://doi.org/10.1145/3637528.3671668.

[101] Junji Jiang, Likang Wu, Hongke Zhao, Hengshu Zhu and Wei Zhang. 'Forecasting movements of stock time series based on hidden state guided deep learning approach'. In: *Information Processing and Management* 60.3 (2023), p. 103328. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2023.103328. URL: https://www.sciencedirect.com/science/article/pii/S0306457323000651.

[102] Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka and Dongjin Song. 'Empowering Time Series Analysis with Large Language Models: A Survey'. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. Ed. by Kate Larson. Survey Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, pp. 8095–8103. DOI: 10.24963/ijcai.2024/895. URL: https://doi.org/10.24963/ijcai.2024/895.

[103] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan and Qingsong Wen. 'Position: What Can Large Language Models Tell Us about Time Series Analysis'. In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. 21–27 Jul 2024, pp. 22260–22276. URL: https://proceedings.mlr.press/v235/jin24i.html.

[104] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick and Ross Girshick. 'CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1988–1997. DOI: 10.1109/CVPR.2017.215.

[105] Michael I. Jordan. 'Attractor dynamics and parallelism in a connectionist sequential machine'. In: *Artificial Neural Networks: Concept Learning*. IEEE Press, 1990, pp. 112–127. ISBN: 0818620153.

[106] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer and Omer Levy. 'SpanBERT: Improving Pre-training by Representing and Predicting Spans'. In: *Transactions of the Association for Computational Linguistics* 8 (2020). Ed. by Mark Johnson, Brian Roark and Ani Nenkova, pp. 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5/.

[107] M. G. Kendall and A. Bradford Hill. 'The Analysis of Economic Time-Series-Part I: Prices'. In: *Journal of the Royal Statistical Society. Series A (General)* 116.1 (1953), pp. 11–34. ISSN: 00359238. URL: http://www.jstor.org/stable/2980947 (visited on 11/02/2023).

[108] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim and Jaewoo Kang. *HATS: A Hierarchical Graph Attention Network for Stock Movement Prediction*. 2019. arXiv: 1908.07999 [q-fin.ST].

[109] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba and Sanja Fidler. 'Skip-Thought Vectors'. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

[110] Kelvin J.L. Koa, Yunshan Ma, Ritchie Ng and Tat-Seng Chua. 'Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction'. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1087–1096. ISBN: 9798400701245. DOI: 10.1145/3583780.3614844. URL: https://doi.org/10.1145/3583780.3614844.

[111] Kelvin J.L. Koa, Yunshan Ma, Ritchie Ng and Tat-Seng Chua. 'Learning to Generate Explainable Stock Predictions using Self-Reflective Large Language Models'. In: *Proceedings of the ACM Web Conference 2024*. Vol. 12706. WWW '24. ACM, May 2024, pp. 4304–4315. DOI: 10.1145/3589334.3645611. URL: http://dx.doi.org/10.1145/3589334.3645611.

[112] Jan Koutnik, Klaus Greff, Faustino Gomez and Juergen Schmidhuber. 'A Clockwork RNN'. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China, 22–24 Jun 2014, pp. 1863–1871. URL: https://proceedings.mlr.press/v32/koutnik14.html.

[113] Litton Jose Kurisinkel, Pruthwik Mishra and Yue Zhang. *Text2TimeSeries: Enhancing Financial Forecasting through Time Series Prediction Updates with Event-Driven Insights from Large Language Models*. 2024. arXiv: 2407.03689 [cs.CL]. URL: https://arxiv.org/abs/2407.03689.

[114] Quoc Le and Tomas Mikolov. 'Distributed Representations of Sentences and Documents'. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China, 22–24 Jun 2014, pp. 1188–1196. URL: https://proceedings.mlr.press/v32/le14.html.

[115] Chengyu Li and Guoqi Qian. 'Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network'. In: *Applied Sciences* 13.1 (2023). ISSN: 2076-3417. DOI: 10.3390/app13010222. URL: https://www.mdpi.com/2076-3417/13/1/222.

[116] Hao Li, Yanyan Shen and Yanmin Zhu. 'Stock Price Prediction Using Attention-based Multi-Input LSTM'. In: *Proceedings of The 10th Asian Conference on Machine Learning*. Ed. by Jun Zhu and Ichiro Takeuchi. Vol. 95. Proceedings of Machine Learning Research. 14–16 Nov 2018, pp. 454–469. URL: https://proceedings.mlr.press/v95/li18c.html.

[117] Haozhou Li, Qinke Peng, Xu Mou, Ying Wang, Zeyuan Zeng and Muhammad Fiaz Bashir. 'Abstractive Financial News Summarization via Transformer-BiLSTM Encoder and Graph Attention-Based Decoder'. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 3190–3205. DOI: 10.1109/TASLP.2023.3304473.

[118] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh D. Gotmare, Shafiq Joty, Caiming Xiong and Steven C.H. Hoi. 'Align before fuse: vision and language representation learning with momentum distillation'. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2021. ISBN: 9781713845393.

[119] Man Li, Ye Zhu, Yuxin Shen and Maia Angelova. 'Clustering-enhanced stock price prediction using deep learning'. In: *World Wide Web* 26.1 (Jan. 2023), pp. 207–232. ISSN: 1573-1413. DOI: 10.1007/s11280-021-01003-0. URL: https://doi.org/10.1007/s11280-021-01003-0.

[120] Menggang Li, Wenrui Li, Fang Wang, Xiaojun Jia and Guangwei Rui. 'Applying BERT to analyze investor sentiment in stock market'. In: *Neural Computing and Applications* 33.10 (May 2021), pp. 4663–4676. ISSN: 1433-3058. DOI: 10.1007/s00521-020-05411-7. URL: https://doi.org/10.1007/s00521-020-05411-7.

[121] Qing Li, Jun Wang, Feng Wang, Ping Li, Ling Liu and Yuanzhu Chen. 'The role of social sentiment in stock markets: a view from joint effects of multiple information sources'. In: *Multimedia Tools and Applications* 76.10 (May 2017), pp. 12315–12345. ISSN: 1573-7721. DOI: 10.1007/s11042-016-3643-4. URL: https://doi.org/10.1007/s11042-016-3643-4.

[122] Shuqi Li, Weiheng Liao, Yuhan Chen and Rui Yan. 'PEN: Prediction-Explanation Network to Forecast Stock Price Movement with Better Explainability'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.4 (June 2023), pp. 5187–5194. DOI: 10.1609/aaai.v37i4.25648. URL: https://ojs.aaai.org/index.php/AAAI/article/view/25648.

[123] Shuqi Li, Yuebo Sun, Yuxin Lin, Xin Gao, Shuo Shang and Rui Yan. 'CausalStock: Deep End-to-end Causal Discovery for News-driven Multi-stock Movement Prediction'. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=5BXXoJh0Vr.

[124] Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen and Sen Huang. 'MASTER: Market-Guided Stock Transformer for Stock Price Forecasting'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.1 (Mar. 2024), pp. 162–170. DOI: 10.1609/aaai.v38i1.27767. URL: https://ojs.aaai.org/index.php/AAAI/article/view/27767.

[125] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu and Qi Su. 'Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction'. In: *Proceedings of the Twenty-Ninth International*

Joint Conference on Artificial Intelligence. Ed. by Christian Bessiere. Special Track on AI in FinTech. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 4541–4547. DOI: 10.24963/ijcai.2020/626. URL: https://doi.org/10.24963/ijcai.2020/626.

[126] Xiangyu Li, Xinjie Shen, Yawen Zeng, Xiaofen Xing and Jin Xu. 'FinReport: Explainable Stock Earnings Forecasting via News Factor Analyzing Model'. In: *Companion Proceedings of the ACM Web Conference 2024*. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 319–327. ISBN: 9798400701726. DOI: 10.1145/3589335.3648330. URL: https://doi.org/10.1145/3589335.3648330.

[127] Xiaohan Li, Jun Wang, Jinghua Tan, Shiyu Ji and Huading Jia. 'A graph neural network-based stock forecasting method utilizing multi-source heterogeneous data fusion'. In: *Multimedia Tools and Applications* 81.30 (Dec. 2022), pp. 43753–43775. ISSN: 1573-7721. DOI: 10.1007/s11042-022-13231-1. URL: https://doi.org/10.1007/s11042-022-13231-1.

[128] Xurui Li et al. 'Heterogeneous Graph Pre-training Based Model for Secure and Efficient Prediction of Default Risk Propagation among Bond Issuers'. In: *Proceedings of the Workshop on Artificial Intelligence System with Confidential Computing*. Network and Distributed System Security Symposium (NDSS). 2024. URL: https://www.ndss-symposium.org/ndss-program/aiscc-2024/.

[129] Yang Li and Yi Pan. 'A novel ensemble deep learning model for stock prediction based on stock prices and news'. In: *International Journal of Data Science and Analytics* 13.2 (Mar. 2022), pp. 139–149. ISSN: 2364-4168. DOI: 10.1007/s41060-021-00279-9. URL: https://doi.org/10.1007/s41060-021-00279-9.

[130] Yanhong Li, Jack Xu and David C. Anastasiu. 'An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*

37.7 (June 2023), pp. 8684–8691. ISSN: 2159-5399. DOI: 10.1609/aaai.v37i7.26045. URL: http://dx.doi.org/10.1609/aaai.v37i7.26045.

[131] Yawei Li, Shuqi Lv, Xinghua Liu and Qiuyue Zhang. 'Incorporating Transformers and Attention Networks for Stock Movement Prediction'. In: *Complexity* 2022 (Feb. 2022), p. 7739087. ISSN: 1076-2787. DOI: 10.1155/2022/7739087. URL: https://doi.org/10.1155/2022/7739087.

[132] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan and Qingsong Wen. 'Foundation Models for Time Series Analysis: A Tutorial and Survey'. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '24. ACM, Aug. 2024, pp. 6555–6565. DOI: 10.1145/3637528.3671451. URL: http://dx.doi.org/10.1145/3637528.3671451.

[133] Kathy Lien. 'Technical Trading Strategy: Multiple Time Frame Analysis'. In: Dec. 2015, pp. 91–100. ISBN: 9781119108412. DOI: 10.1002/9781119212997.ch8.

[134] Hengxu Lin, Dong Zhou, Weiqing Liu and Jiang Bian. 'Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport'. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 1017–1026. ISBN: 9781450383325. DOI: 10.1145/3447548.3467358. URL: https://doi.org/10.1145/3447548.3467358.

[135] Lorenz Linhardt, Klaus-Robert Müller and Grégoire Montavon. 'Preemptively pruning Clever-Hans strategies in deep neural networks'. In: *Information Fusion* 103 (2024), p. 102094. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.102094. URL: https://www.sciencedirect.com/science/article/pii/S1566253523004104.

[136] Chang Liu, Jie Yan, Feiyue Guo and Min Guo. 'Forecasting the Market with Machine Learning Algorithms: An Application of NMC-BERT-LSTM-DQN-X Algorithm in Quantitative Trading'. In: *ACM Transactions on Knowledge Discovery from Data* 16.4 (Jan. 2022). ISSN: 1556-4681. DOI: 10.1145/3488378. URL: https://doi.org/10.1145/3488378.

[137] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li and Rui Zhao. 'Spatial-Temporal Large Language Model for Traffic Prediction'. In: *Proceedings of the 25th IEEE International Conference on Mobile Data Management*. 2024, pp. 31–40. DOI: 10.1109/MDM55031.2024.00015.

[138] Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao and Liang Yang. 'Transformer-Based Capsule Network For Stock Movement Prediction'. In: *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*. Macao, China, Aug. 2019, pp. 66–73. URL: https://aclanthology.org/W19-5511.

[139] Mengpu Liu, Mengying Zhu, Xiuyuan Wang, Guofang Ma, Jianwei Yin and Xiaolin Zheng. 'ECHO-GL: Earnings Calls-Driven Heterogeneous Graph Learning for Stock Movement Prediction'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.12 (Mar. 2024), pp. 13972–13980. DOI: 10.1609/aaai.v38i12.29305. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29305.

[140] Yang Liu. 'Fine-tune BERT for Extractive Summarization'. In: *arXiv preprint* abs/1903.10318 (2019). arXiv: 1903.10318. URL: http://arxiv.org/abs/1903.10318.

[141] Yong Liu, Haixu Wu, Jianmin Wang and Mingsheng Long. 'Non-stationary transformers: exploring the stationarity in time series forecasting'. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[142] Yuxin Liu, Jimin Lin and Achintya Gopal. *NeuralBeta: Estimating Beta Using Deep Learning*. 2024. arXiv: 2408.01387 [q-fin.ST]. URL: https://arxiv.org/abs/2408.01387.

[143] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun and Jingyang Wang. 'A CNN-LSTM-Based Model to Forecast Stock Prices'. In: *Complexity* 2020 (Nov. 2020), p. 6622927. ISSN: 1076-2787. DOI: 10.1155/2020/6622927. URL: https://doi.org/10.1155/2020/6622927.

[144] Wenjie Lu, Jiazheng Li, Jingyang Wang and Lele Qin. 'A CNN-BiLSTM-AM method for stock price prediction'. In: *Neural Computing and Applications* 33.10 (May 2021), pp. 4741–4753. ISSN: 1433-3058. DOI: 10.1007/s00521-020-05532-z. URL: https://doi.org/10.1007/s00521-020-05532-z.

[145] Chang Luo, Tiejun Ma and Mihai Cucuringu. 'Spatial-Temporal Stock Movement Prediction and Portfolio Selection based on the Semantic Company Relationship Graph'. In: (2024). Available at SSRN: https://ssrn.com/abstract=4699179 or http://dx.doi.org/10.2139/ssrn.4699179.

[146] Di Luo, Weiheng Liao, Shuqi Li, Xin Cheng and Rui Yan. 'Causality-Guided Multi-Memory Interaction Network for Multivariate Stock Price Movement Prediction'. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 12164–12176. DOI: 10.18653/v1/2023.acl-long.679. URL: https://aclanthology.org/2023.acl-long.679.

[147] Yongen Luo, Jicheng Hu, Xiaofeng Wei, Dongjian Fang and Heng Shao. 'Stock trends prediction based on hypergraph modeling clustering algorithm'. In: *2014 IEEE International Conference on Progress in Informatics and Computing*. 2014, pp. 27–31. DOI: 10.1109/PIC.2014.6972289.

[148] Ronny Luss and Alexandre d'Aspremont. 'Support Vector Machine Classification with Indefinite Kernels'. In: *Mathematical Programming Computation* 1.2-3 (2009), pp. 97–118. DOI: 10.1007/s12532-009-0005-5.

[149] Ye Ma, Lu Zong, Yikang Yang and Jionglong Su. 'News2vec: News Network Embedding with Subnode Information'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4843–4852. DOI: 10.18653/v1/D19-1490. URL: https://aclanthology.org/D19-1490.

[150] Muhammad Anwar Ma'sum, MD Rasel Sarkar, Mahardhika Pratama, Savitha Ramasamy, Sreenatha Anavatti, Lin Liu, Habibullah Habibullah and Ryszard Kowalczyk. 'Dynamic Long-Term Time-Series Forecasting via Meta Transformer Networks'. In: *IEEE Transactions on Artificial Intelligence* 5.8 (2024), pp. 4258–4268. DOI: 10.1109/TAI.2024.3365775.

[151] Xiliu Man, Jianwu Lin and Yujiu Yang. 'Stock-UniBERT: A News-based Cost-sensitive Ensemble BERT Model for Stock Trading'. In: *2020 IEEE 18th International Conference on Industrial Informatics*. Vol. 1. 2020, pp. 440–445. DOI: 10.1109/INDIN45582.2020.9442147.

[152] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN: 9780521865715.

[153] Harry Markowitz. 'Portfolio Selection'. In: *The Journal of Finance* 7.1 (1952), pp. 77–91. DOI: 10.2307/2975974. URL: https://doi.org/10.2307/2975974.

[154] Sidra Mehtab and Jaydip Sen. 'Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models'. In: *2020 International Conference on Decision Aid Sciences and Application*. 2020, pp. 447–453. DOI: 10.1109/DASA51403.2020.9317207.

[155] Attilio Meucci. 'The Black-Litterman Approach: Original Model and Extensions'. In: *The Encyclopedia of Quantitative Finance* (Aug. 2008). Shorter version published in 2010. DOI: 10.2139/ssrn.1117574. URL: https://ssrn.com/abstract=1117574.

[156] Tomás Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 'Efficient Estimation of Word Representations in Vector Space'. In: *1st International Conference on Learning Representations, 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. 2013. URL: http://arxiv.org/abs/1301.3781.

[157] Felix Prenzel Milena Vuletić and Mihai Cucuringu. 'Fin-GAN: forecasting and classifying financial time series via generative adversarial networks'. In: *Quantitative Finance* 24.2 (2024), pp. 175–199. DOI: 10.1080/14697688.2023.2299466.

[158] Latrisha N. Mintarya, Jeta N.M. Halim, Callista Angie, Said Achmad and Aditya Kurniawan. 'Machine learning approaches in stock market prediction: A systematic literature review'. In: 216 (2023). 7th International Conference on Computer Science and Computational Intelligence 2022, pp. 96–102. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.12.115. URL: https://www.sciencedirect.com/science/article/pii/S1877050922021937.

[159] Seyed Mirjebreili, Ata Solouki, Hamidreza Soltanalizadeh and Mohammad Sabokrou. 'Multi-Task Transformer for Stock Market Trend Prediction'. In: Nov. 2022, pp. 101–105. DOI: 10.1109/ICCKE57176.2022.9960122.

[160] Nikhil Muralidhar, Mohammad Islam, Manish Marwah, Anuj Karpatne and Naren Ramakrishnan. 'Incorporating Prior Domain Knowledge into Deep Neural Networks'. In: Dec. 2018, pp. 36–45. DOI: 10.1109/BigData.2018.8621955.

[161] Williamson Murray and Robert H. Scales. 'The Iraq War: A Military History'. In: Harvard University Press, 2003. ISBN: 9780674012806. URL:

http://www.jstor.org/stable/j.ctvjf9wjs.6 (visited on 12/04/2024).

[162] David M. Q. Nelson, Adriano C. M. Pereira and Renato A. de Oliveira. 'Stock market's price movement prediction with LSTM neural networks'. In: *2017 International Joint Conference on Neural Networks*. 2017, pp. 1419–1426. DOI: 10.1109/IJCNN.2017.7966019.

[163] Thi-Thu Nguyen and Seokhoon Yoon. 'A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning'. In: *Applied Sciences* 9.22 (2019). ISSN: 2076-3417. DOI: 10.3390/app9224745. URL: https://www.mdpi.com/2076-3417/9/22/4745.

[164] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen and Stefan Zohren. *A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges*. 2024. arXiv: 2406.11903 [q-fin.GN]. URL: https://arxiv.org/abs/2406.11903.

[165] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong and Jayant Kalagnanam. 'A Time Series is Worth 64 Words: Long-term Forecasting with Transformers'. In: *Proceedings of the 11th International Conference on Learning Representations*. 2023. URL: https://arxiv.org/abs/2211.14730.

[166] Hao Niu, Yun Xiong, Xiaosu Wang, Wenjing Yu, Yao Zhang and Weizu Yang. 'KeFVP: Knowledge-enhanced Financial Volatility Prediction'. In: *Findings of the Association for Computational Linguistics: 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 11499–11513. DOI: 10.18653/v1/2023.findings-emnlp.770. URL: https://aclanthology.org/2023.findings-emnlp.770.

[167] Yingjie Niu, Lanxin Lu, Rian Dolphin, Valerio Poti and Ruihai Dong. 'Evaluating Financial Relational Graphs: Interpretation Before Prediction'. In: *Proceedings of the 5th ACM International Conference on AI in Finance*.

Brooklyn, NY, USA: Association for Computing Machinery, 2024, pp. 564–572. DOI: 10.1145/3677052.3698644. URL: https://doi.org/10.1145/3677052.3698644.

[168] Paraskevi Nousi, Loukia Avramelou, Georgios Rodinos, Maria Tzelepi, Theodoros Manousis, Konstantinos Tsampazis, Kyriakos Stefanidis, Dimitris Spanos, Manos Kirtas, Pavlos Tosidis, Avraam Tsantekidis, Nikolaos Passalis and Anastasios Tefas. *Leveraging Deep Learning and Online Source Sentiment for Financial Portfolio Management*. 2023. arXiv: 2309.16679 [q-fin.PM]. URL: https://arxiv.org/abs/2309.16679.

[169] Kenniy Olorunnimbe and Herna Viktor. 'Towards efficient similarity embedded temporal Transformers via extended timeframe analysis'. In: *Complex & Intelligent Systems* 10.4 (Aug. 2024), pp. 4793–4815. ISSN: 2198-6053. DOI: 10.1007/s40747-024-01400-8. URL: https://doi.org/10.1007/s40747-024-01400-8.

[170] Soroush Omranpour, Guillaume Rabusseau and Reihaneh Rabbany. *Higher Order Transformers: Enhancing Stock Movement Prediction On Multimodal Time-Series Data*. 2024. arXiv: 2412.10540 [cs.LG]. URL: https://arxiv.org/abs/2412.10540.

[171] Santiago Ontanon, Joshua Ainslie, Zachary Fisher and Vaclav Cvicek. 'Making Transformers Solve Compositional Tasks'. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3591–3607. DOI: 10.18653/v1/2022.acl-long.251. URL: https://aclanthology.org/2022.acl-long.251.

[172] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior and Koray Kavukcuoglu. 'WaveNet: A Generative Model for Raw Audio'. In: *Proceedings of the 9th ISCA Speech Synthesis Workshop*. 2016, p. 125.

[173] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[174] Bowen Pang, Wei Wei, Xing Li, Xiangnan Feng and Chao Li. 'A representation-learning-based approach to predict stock price trend via dynamic spatiotemporal feature embedding'. In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 106849. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.106849. URL: https://www.sciencedirect.com/science/article/pii/S0952197623010333.

[175] Manali Patel, Krupa Jariwala and Chiranjoy Chattopadhyay. 'A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting'. In: *ACM Computing Surveys* 57.2 (Oct. 2024). ISSN: 0360-0300. DOI: 10.1145/3696411. URL: https://doi.org/10.1145/3696411.

[176] Ramkrishna Patel, Vikas Choudhary, Deepika Saxena and Ashutosh Kumar Singh. 'Review of stock prediction using machine learning techniques'. In: *2021 5th International Conference on Trends in Electronics and Informatics*. IEEE. 2021, pp. 840–846.

[177] Yunhua Pei, Jin Zheng and John Cartlidge. 'Dynamic Graph Representation with Contrastive Learning for Financial Market Prediction: Integrating Temporal Evolution and Static Relations'. In: *Proceedings of the 17th International Conference on Agents and Artificial Intelligence*. Association for Computing Machinery, 2025, pp. 298–309. DOI: 10.5220/0013154700003890. URL: https://arxiv.org/abs/2412.04034.

[178] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak,

Zhenyuan Zhang, Qinghua Zhou, Jian Zhu and Rui-Jie Zhu. ‘RWKV: Reinventing RNNs for the Transformer Era’. In: *Findings of the Association for Computational Linguistics: 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 14048–14077. DOI: 10.18653/v1/2023.findings-emnlp.936. URL: https://aclanthology.org/2023.findings-emnlp.936.

[179] Hao Peng, Ke Dong and Jie Yang. ‘Stock Price Movement Prediction based on Relation Type guided Graph Convolutional Network’. In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 106948. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.106948. URL: https://www.sciencedirect.com/science/article/pii/S0952197623011326.

[180] Peng Peng, Yuehong Chen, Weiwei Lin and James Z. Wang. ‘Attention-based CNN–LSTM for high-frequency multiple cryptocurrency trend prediction’. In: *Expert Systems with Applications* 237 (2024), p. 121520. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121520. URL: https://www.sciencedirect.com/science/article/pii/S0957417423020225.

[181] Jeffrey Pennington, Richard Socher and Christopher D. Manning. ‘GloVe: Global Vectors for Word Representation’. In: *Empirical Methods in Natural Language Processing*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[182] Bryan Perozzi, Rami Al-Rfou and Steven Skiena. ‘DeepWalk: online learning of social representations’. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. Vol. 2. KDD ’14. ACM, Aug. 2014, pp. 701–710. DOI: 10.1145/2623330.2623732. URL: http://dx.doi.org/10.1145/2623330.2623732.

[183] Sarah Perrin and Thierry Roncalli. ‘Machine Learning Optimization Algorithms & Portfolio Allocation’. In: *Machine Learning for Asset*

Management. Wiley, 2020. Chap. 8, pp. 261–328. DOI: 10.1002/9781119751182.ch8.

[184] E. Prasetyo and K. D. Hartomo. 'Multi-industry stock forecasting using GRU-LSTM deep transfer learning method'. In: *INFOTEL* 15.2 (May 2023), pp. 150–163.

[185] Hao Qian, Hongting Zhou, Qian Zhao, Hao Chen, Hongxiang Yao, Jingwei Wang, Ziqi Liu, Fei Yu, Zhiqiang Zhang and Jun Zhou. 'MDGNN: multi-relational dynamic graph neural network for comprehensive and dynamic stock investment prediction'. In: AAAI Press, 2024. ISBN: 978-1-57735-887-9. DOI: 10.1609/aaai.v38i13.29381. URL: https://doi.org/10.1609/aaai.v38i13.29381.

[186] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang and Garrison W. Cottrell. 'A dual-stage attention-based recurrent neural network for time series prediction'. In: Melbourne, Australia: AAAI Press, 2017, pp. 2627–2633. ISBN: 9780999241103.

[187] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. 'Language Models are Unsupervised Multitask Learners'. In: (2019).

[188] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu. 'Exploring the limits of transfer learning with a unified text-to-text transformer'. In: *Journal of Machine Learning Research* 21.1 (Jan. 2020). ISSN: 1532-4435.

[189] Eduardo Ramos-Pérez, Pablo J. Alonso-González and José Javier Núñez-Velázquez. 'Multi-Transformer: A New Neural Network-Based Architecture for Forecasting S and P Volatility'. In: *Mathematics* 9.15 (2021). ISSN: 2227-7390. DOI: 10.3390/math9151794. URL: https://www.mdpi.com/2227-7390/9/15/1794.

[190] Jawad Rasheed, Akhtar Jamil, Alaa Ali Hameed, Muhammad Ilyas, Adem Özyavaş and Naim Ajlouni. 'Improving Stock Prediction Accuracy Using CNN and LSTM'. In: *2020 International Conference on Data Analytics for*

Business and Industry: Way Towards a Sustainable Economy. 2020, pp. 1–5. DOI: 10.1109/ICDABI51230.2020.9325597.

[191] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka and Irina Rish. 'Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting'. In: *Proceedings of the R0-FoMo Workshop at the Conference on Neural Information Processing Systems 2023*. 2023. URL: https://arxiv.org/abs/2310.08278.

[192] Akhter Mohiuddin Rather, Arun Agarwal and V.N. Sastry. 'Recurrent neural network and a hybrid model for prediction of stock returns'. In: *Expert Systems with Applications* 42.6 (2015), pp. 3234–3241. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0957417414007684.

[193] Paramita Ray, Bhaswati Ganguli and Amlan Chakrabarti. 'A Hybrid Approach of Bayesian Structural Time Series With LSTM to Identify the Influence of News Sentiment on Short-Term Forecasting of Stock Price'. In: *IEEE Transactions on Computational Social Systems* 8.5 (2021), pp. 1153–1162. DOI: 10.1109/TCSS.2021.3073964.

[194] Nils Reimers and Iryna Gurevych. 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410/.

[195] Anna Rogers, Aleksandr Drozd and Bofang Li. 'The (too Many) Problems of Analogical Reasoning with Word Vectors'. In: *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 135–148. DOI: 10.18653/v1/S17-1017. URL: https://aclanthology.org/S17-1017.

[196] Bhaskarjit Sarmah, Nayana Nair, Riya Jain, Dhagash Mehta and Stefano Pasquali. 'Learning Embedded Representation of the Stock Correlation Matrix Using Graph Machine Learning'. In: *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics, Hoboken, NJ, USA, October 22-23, 2024*. IEEE, 2024, pp. 1–9. DOI: 10.1109/CIFER62890.2024.10772849. URL: https://doi.org/10.1109/CIFEr62890.2024.10772849.

[197] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr and Rajiv Ratn Shah. 'Stock Selection via Spatiotemporal Hypergraph Attention Network: A Learning to Rank Approach'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.1 (May 2021), pp. 497–504. DOI: 10.1609/aaai.v35i1.16127. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16127.

[198] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa and Rajiv Shah. 'Exploring the Scale-Free Nature of Stock Markets: Hyperbolic Graph Learning for Algorithmic Trading'. In: *Proceedings of the Web Conference 2021*. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 11–22. ISBN: 9781450383127. DOI: 10.1145/3442381.3450095. URL: https://doi.org/10.1145/3442381.3450095.

[199] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa and Rajiv Ratn Shah. 'Deep Attentive Learning for Stock Movement Prediction From Social Media Text and Company Correlations'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 8415–8426. DOI: 10.18653/

v1/2020.emnlp-main.676. URL: https://aclanthology.org/2020.emnlp-main.676.

[200] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa and Rajiv Ratn Shah. 'Spatiotemporal Hypergraph Convolution Network for Stock Movement Forecasting'. In: *2020 IEEE International Conference on Data Mining*. 2020, pp. 482–491. DOI: 10.1109/ICDM50108.2020.00057.

[201] Ramit Sawhney, Arnav Wadhwa, Shivam Agarwal and Rajiv Ratn Shah. 'FAST: Financial News and Tweet Based Time Aware Network for Stock Trading'. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann and Reut Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 2164–2175. DOI: 10.18653/v1/2021.eacl-main.185. URL: https://aclanthology.org/2021.eacl-main.185.

[202] Ramit Sawhney, Arnav Wadhwa, Shivam Agarwal and Rajiv Ratn Shah. 'Quantitative Day Trading from Natural Language using Reinforcement Learning'. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty and Yichao Zhou. Online: Association for Computational Linguistics, June 2021, pp. 4018–4030. DOI: 10.18653/v1/2021.naacl-main.316. URL: https://aclanthology.org/2021.naacl-main.316.

[203] Kanghyeon Seo, Seungjae Lee, Woo Jin Cho, Yoojeong Song and Jihoon Yang. 'Multi-time Window Ensemble and Maximization of Expected Return for Stock Movement Prediction'. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by De-Nian Yang, Xing Xie, Vincent S. Tseng, Jian Pei, Jen-Wei Huang and Jerry Chun-Wei Lin. Singapore: Springer Nature Singapore, 2024, pp. 17–29. ISBN: 978-981-97-2238-9.

[204] Kanghyeon Seo and Jihoon Yang. 'Exploring The Efficient Market Hypothesis for Accurate Stock Movement Prediction via Feature-Axis Transformer'. In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. Avila, Spain: Association for Computing Machinery, 2024, pp. 892–901. ISBN: 9798400702433. DOI: 10.1145/3605098.3635928. URL: https://doi.org/10.1145/3605098.3635928.

[205] Peter Shaw, Jakob Uszkoreit and Ashish Vaswani. 'Self-Attention with Relative Position Representations'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 464–468. DOI: 10.18653/v1/N18-2074. URL: https://aclanthology.org/N18-2074.

[206] Yang Shen, Jicheng Hu, Yanan Lu and Xiaofeng Wang. 'Stock trends prediction by hypergraph modeling'. In: *2012 IEEE International Conference on Computer Science and Automation Engineering*. 2012, pp. 104–107. DOI: 10.1109/ICSESS.2012.6269415.

[207] Zhuangwei Shi. *MambaStock: Selective state space model for stock prediction*. 2024. arXiv: 2402.18959 [cs.CE]. URL: https://arxiv.org/abs/2402.18959.

[208] Ahyun Song, Euiseong Seo and Heeyoul Kim. 'Anomaly VAE-Transformer: A Deep Learning Approach for Anomaly Detection in Decentralized Finance'. In: *IEEE Access* 11 (2023), pp. 98115–98131. DOI: 10.1109/ACCESS.2023.3313448.

[209] Chen-Hui Song, Xi Xiao, Bin Zhang and Shu-Tao Xia. 'Follow the Will of the Market: A Context-Informed Drift-Aware Method for Stock Prediction'. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. Birmingham, United Kingdom: Association for Computing Machinery, 2023, pp. 2311–2320. ISBN: 9798400701245. DOI: 10.1145/3583780.3614886. URL: https://doi.org/10.1145/3583780.3614886.

[210] Priyank Sonkiya, Vikas Bajpai and Anukriti Bansal. *Stock price prediction using BERT and GAN*. 2021. arXiv: 2107.09055 [q-fin.ST].

[211] Troy J. Strader, John J. Rozycki, Thomas H. Root and Yu-Hsiang Huang. 'Machine Learning Stock Market Prediction Studies: Review and Research Directions'. In: *Journal of International Technology and Information Management* (2020).

[212] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo and Yunfeng Liu. 'RoFormer: Enhanced transformer with Rotary Position Embedding'. In: *Neurocomputing* 568.C (Feb. 2024). ISSN: 0925-2312. DOI: 10.1016/j.neucom.2023.127063. URL: https://doi.org/10.1016/j.neucom.2023.127063.

[213] Shuo Sun, Xinrun Wang, Wanqi Xue, Xiaoxuan Lou and Bo An. 'Mastering Stock Markets with Efficient Mixture of Diversified Trading Experts'. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Long Beach, CA, USA: Association for Computing Machinery, 2023, pp. 2109–2119. ISBN: 9798400701030. DOI: 10.1145/3580305.3599424. URL: https://doi.org/10.1145/3580305.3599424.

[214] Z. Tang, J. Huang and D. Rinprasertmeechai. 'Period-aggregated transformer for learning latent seasonalities in long-horizon financial time series'. In: *PLoS One* 19.8 (2024). Published 2024 Aug 8, e0308488. DOI: 10.1371/journal.pone.0308488. URL: https://doi.org/10.1371/journal.pone.0308488.

[215] Hanshuang Tong, Jun Li, Ning Wu, Ming Gong, Dongmei Zhang and Qi Zhang. *Ploutos: Towards interpretable stock movement prediction with financial large language model*. Papers 2403.00782. arXiv.org, Feb. 2024. URL: https://ideas.repec.org/p/arx/papers/2403.00782.html.

[216] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave

and Guillaume Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[217] Dimitrios Vamvourellis, Máté Tóth, Snigdha Bhagat, Dhruv Desai, Dhagash Mehta and Stefano Pasquali. 'Company Similarity Using Large Language Models'. In: *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics, Hoboken, NJ, USA, October 22-23, 2024*. IEEE, 2024, pp. 1–9. DOI: 10.1109/CIFER62890.2024.10772990. URL: https://doi.org/10.1109/CIFEr62890.2024.10772990.

[218] Uras Varolgunes, Shibo Yao, Yao Ma and Dantong Yu. 'Embedding Imputation With Self-Supervised Graph Neural Networks'. In: *IEEE Access* 11 (2023), pp. 70610–70620. DOI: 10.1109/ACCESS.2023.3292314.

[219] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser and Illia Polosukhin. 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[220] Frederic Voigt. *Adapting Natural Language Processing Strategies for Stock Price Prediction*. DC@KI2023: Proceedings of Doctoral Consortium at KI 2023. 2023. DOI: 10.18420/ki2023-dc-03.

[221] Frederic Voigt, Jose Alcaraz Calero, Keshav Dahal, Qi Wang, Kai von Luck and Peer Stelldinger. 'Towards Machine Learning Based Text Categorization in the Financial Domain'. In: *2024 IEEE 3rd Conference on Information Technology and Data Science*. 2024, pp. 1–6. DOI: 10.1109/CITDS62610.2024.10791384.

[222] Frederic Voigt, Jose Alcaraz Calero, Keshav Dahal, Qi Wang, Kai von Luck and Peer Stelldinger. 'Adapting Speech Models for Stock Price Prediction'. In: *2024 IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications*. 2024, pp. 1–8. DOI: 10.1109/ICCCMLA63077.2024.10871633.

[223] Frederic Voigt, Jose Alcaraz Calero, Keshav P. Dahal, Qi Wang, Kai von Luck and Peer Stelldinger. 'Quantitative Market Situation Embeddings: Utilizing Doc2Vec Strategies for Stock Data'. In: *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics, Hoboken, NJ, USA, October 22-23, 2024*. IEEE, 2024, pp. 1–10. DOI: 10.1109/CIFER62890.2024.10772772. URL: https://doi.org/10.1109/CIFEr62890.2024.10772772.

[224] Frederic Voigt, Kai von Luck and Peer Stelldinger. 'Assessment of the Applicability of Large Language Models for Quantitative Stock Price Prediction'. In: *Proceedings of the 17th International Conference on Pervasive Technologies Related to Assistive Environments*. Crete, Greece: Association for Computing Machinery, 2024, pp. 293–302. ISBN: 9798400717604. DOI: 10.1145/3652037.3652047. URL: https://doi.org/10.1145/3652037.3652047.

[225] Ahmed. S. Wafi, Hassan Hassan and Adel Mabrouk. 'Fundamental Analysis Models in Financial Markets – Review Study'. In: *Procedia Economics and Finance* 30 (2015). 3rd and 4th Economics and Finance Conference, pp. 939–947. ISSN: 2212-5671. DOI: https://doi.org/10.1016/S2212-5671(15)01344-1. URL: https://www.sciencedirect.com/science/article/pii/S2212567115013441.

[226] Qizhi Wan, Changxuan Wan, Keli Xiao, Dexi Liu, Chenliang Li, Bolong Zheng, Xiping Liu and Rong Hu. 'Joint Document-Level Event Extraction via Token-Token Bidirectional Event Completed Graph'. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber and

Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 10481–10492. DOI: 10.18653/v1/2023.acl-long.584. URL: https://aclanthology.org/2023.acl-long.584.

[227] Bin Wang, Angela Wang, Fenxiao Chen, Yun Cheng Wang and C.-C. Jay Kuo. 'Evaluating Word Embedding Models: Methods and Experimental Results'. In: *APSIPA Transactions on Signal and Information Processing* 8 (2019).

[228] Haiyao Wang, Jianxuan Wang, Lihui Cao, Yifan Li, Qiuhong Sun and Jingyang Wang. 'A Stock Closing Price Prediction Model Based on CNN-BiSLSTM'. In: *Complexity* 2021 (Sept. 2021), p. 5360828. ISSN: 1076-2787. DOI: 10.1155/2021/5360828. URL: https://doi.org/10.1155/2021/5360828.

[229] Heyuan Wang, Shun Li, Tengjiao Wang and Jiayi Zheng. 'Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction'. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Ed. by Zhi-Hua Zhou. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 3691–3698. DOI: 10.24963/ijcai.2021/508. URL: https://doi.org/10.24963/ijcai.2021/508.

[230] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Weijun Chen and Wei Chen. 'Agree to Disagree: Personalized Temporal Embedding and Routing for Stock Forecast'. In: *IEEE Transactions on Knowledge and Data Engineering* 36.9 (2024), pp. 4398–4410. DOI: 10.1109/TKDE.2024.3374373.

[231] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan and Wei Chen. 'Adaptive Long-Short Pattern Transformer for Stock Investment Selection'. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 3970–3977. DOI: 10.24963/ijcai.2022/551. URL: https://doi.org/10.24963/ijcai.2022/551.

[232] Heyuan Wang, Tengjiao Wang and Yi Li. 'Incorporating Expert-Based Investment Opinion Signals in Stock Prediction: A Deep Learning Framework'. In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020*. 2020. URL: https://api.semanticscholar.org/CorpusID:214262181.

[233] Jia Wang, Tong Sun, Benyuan Liu, Yu Cao and Hongwei Zhu. 'CLVSA: A Convolutional LSTM Based Variational Sequence-to-Sequence Model with Attention for Predicting Trends of Financial Markets'. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. IJCAI-2019. International Joint Conferences on Artificial Intelligence Organization, Aug. 2019. DOI: 10.24963/ijcai.2019/514. URL: http://dx.doi.org/10.24963/ijcai.2019/514.

[234] Shengkun Wang, Taoran Ji, Linhan Wang, Yanshen Sun, Shang-Ching Liu, Amit Kumar and Chang-Tien Lu. *StockTime: A Time Series Specialized Large Language Model Architecture for Stock Price Prediction*. 2024. arXiv preprint arXiv:2409.08281: 2409.08281. URL: https://arxiv.org/abs/2409.08281.

[235] Aaron Wheeler and Jeffrey D. Varner. *MarketGPT: Developing a Pretrained transformer (GPT) for Modeling Financial Time Series*. Papers arXiv preprint arXiv:2411.16585. arXiv.org, Nov. 2024. URL: https://ideas.repec.org/p/arx/papers/2411.16585.html.

[236] Kieran Wood, Sven Giegerich, Stephen Roberts and Stefan Zohren. 'Trading with the Momentum Transformer: An Intelligent and Interpretable Architecture'. In: *Risk* (Mar. 2023). URL: https://www.risk.net/cutting-edge/7956074/trading-with-the-momentum-transformer-an-interpretable-deep-learning-architecture.

[237] Jheng-Long Wu, Xian-Rong Tang and Chin-Hsiung Hsu. 'A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches'. In: *Soft Computing* 27.12 (June 2023), pp. 8209–8222. ISSN: 1433-7479. DOI: 10.1007/s00500-

022-07716-2. URL: https://doi.org/10.1007/s00500-022-07716-2.

[238] Jimmy Ming-Tai Wu, Zhongcui Li, Norbert Herencsar, Bay Vo and Jerry Chun-Wei Lin. 'A graph-based CNN-LSTM stock price prediction algorithm with leading indicators'. In: *Multimedia Systems* 29.3 (June 2023), pp. 1751–1770. ISSN: 1432-1882. DOI: 10.1007/s00530-021-00758-w. URL: https://doi.org/10.1007/s00530-021-00758-w.

[239] Jimmy Ming-Tai Wu, Zhongcui Li, Gautam Srivastava, Jaroslav Frnda, Vicente Garcia Diaz and Jerry Chun-Wei Lin. 'A CNN-based Stock Price Trend Prediction with Futures and Historical Price'. In: *2020 International Conference on Pervasive Artificial Intelligence*. 2020, pp. 134–139. DOI: 10.1109/ICPAI51961.2020.00032.

[240] Mei-Chen Wu, Szu-Hao Huang and An-Pin Chen. 'Momentum portfolio selection based on learning-to-rank algorithms with heterogeneous knowledge graphs'. In: *Applied Intelligence* 54.5 (Mar. 2024), pp. 4189–4209. ISSN: 1573-7497. DOI: 10.1007/s10489-024-05377-2. URL: https://doi.org/10.1007/s10489-024-05377-2.

[241] Shengting Wu, Yuling Liu, Ziran Zou and Tien-Hsiung Weng. 'S I LSTM: stock price prediction based on multiple data sources and sentiment analysis'. In: *Connection Science* 34 (June 2021), pp. 1–19. DOI: 10.1080/09540091.2021.1940101.

[242] Hongjie Xia, Huijie Ao, Long Li, Yu Liu, Sen Liu, Guangnan Ye and Hongfeng Chai. 'CI-STHPAN: Pre-trained Attention Network for Stock Selection with Channel-Independent Spatio-Temporal Hypergraph'. In: *Proceedings of the Thirty-Eighth Conference on Artificial Intelligence* 38.8 (Mar. 2024), pp. 9187–9195. DOI: 10.1609/aaai.v38i8.28770. URL: https://ojs.aaai.org/index.php/AAAI/article/view/28770.

[243] Yongqin Xian, Christoph H. Lampert, Bernt Schiele and Zeynep Akata. ' Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad

and the Ugly '. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.09 (Sept. 2019), pp. 2251–2265. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2018.2857768. URL: https://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2857768.

[244] Cong Xu, Huiling Huang, Xiaoting Ying, Jianliang Gao, Zhao Li, Peng Zhang, Jie Xiao, Jiarun Zhang and Jiangjian Luo. 'HGNN: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks'. In: *Information Sciences* 607 (2022), pp. 783–798. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.06.010. URL: https://www.sciencedirect.com/science/article/pii/S0020025522005928.

[245] Jia Xu and Longbing Cao. 'Copula Variational LSTM for High-Dimensional Cross-Market Multivariate Dependence Modeling'. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023), pp. 1–15. DOI: 10.1109/TNNLS.2023.3293131.

[246] Yuanjian Xu, Anxian Liu, Jianing Hao, Zhenzhuo Li, Shichang Meng and Guang Zhang. *PLUTUS: A Well Pre-trained Large Unified Transformer can Unveil Financial Time Series Regularities*. 2024. arXiv: 2408.10111. URL: https://arxiv.org/abs/2408.10111.

[247] Yumo Xu and Shay B. Cohen. 'Stock Movement Prediction from Tweets and Historical Prices'. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1970–1979. DOI: 10.18653/v1/P18-1183. URL: https://aclanthology.org/P18-1183.

[248] Hao Xue and Flora D. Salim. 'PromptCast: A New Prompt-Based Learning Paradigm for Time Series Forecasting'. In: *IEEE Transactions on Knowledge and Data Engineering* 36.11 (2024), pp. 6851–6864. DOI: 10.1109/TKDE.2023.3342137.

[249] Wenbo Yan and Ying Tan. *TCGPN: Temporal-Correlation Graph Pre-trained Network for Stock Forecasting*. Tech. rep. 2407.18519. arXiv.org, July 2024. URL: https://ideas.repec.org/p/arx/papers/2407.18519.html.

[250] Linyi Yang, James Ng, Barry Smyth and Ruihai Dong. 'HTML: Hierarchical Transformer-based Multi-task Learning for Volatility Prediction'. In: *Proceedings of The Web Conference 2020*. New York, NY, USA: Association for Computing Machinery, Apr. 2020. DOI: 10.1145/3366423.3380128. URL: https://doi.org/10.1145/3366423.3380128.

[251] Linyi Yang, Zheng Zhang, Su Xiong, Lirui Wei, James Ng, Lina Xu and Ruihai Dong. 'Explainable Text-Driven Neural Network for Stock Prediction'. In: *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems*. 2018, pp. 441–445. DOI: 10.1109/CCIS.2018.8691233.

[252] Yi Yang, Mark Christopher Siy Uy and Allen Huang. 'FinBERT: A Pretrained Language Model for Financial Communications'. In: *CoRR* abs/2006.08097 (2020). arXiv preprint: 2006.08097. URL: https://arxiv.org/abs/2006.08097.

[253] Zhen Yang, Tianlong Zhao, Suwei Wang and Xuemei Li. 'MDF-DMC: A stock prediction model combining multi-view stock data features with dynamic market correlation information'. In: *Expert Systems with Applications* 238 (2024), p. 122134. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122134. URL: https://www.sciencedirect.com/science/article/pii/S0957417423026362.

[254] Xingcheng Yao, Yanan Zheng, Xiaocong Yang and Zhilin Yang. 'NLP From Scratch Without Large-Scale Pretraining: A Simple and Efficient Framework'. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu and Sivan Sabato. Vol. 162. Proceedings of Machine

Learning Research. PMLR, 17–23 Jul 2022, pp. 25438–25451. URL: https://proceedings.mlr.press/v162/yao22c.html.

[255] Ziruo Yi, Ting Xiao, Ijeoma Kaz-Onyeakazi, Cheran Ratnam, Theophilus Medeiros, Phillip Nelson et al. *Stock2Vec: An Embedding to Improve Predictive Models for Companies*. https://digital.library.unt.edu/ark:/67531/metadc2047084/. Accessed April 8, 2025. University of North Texas Libraries, UNT Digital Library. Crediting UNT College of Information. June 2022.

[256] Xingkun Yin, Da Yan, Abdullateef Almudaifer, Sibo Yan and Yang Zhou. 'Forecasting Stock Prices Using Stock Correlation Graph: A Graph Convolutional Network Approach'. In: *2021 International Joint Conference on Neural Networks*. 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9533510.

[257] Zelin Ying, Dawei Cheng, Cen Chen, Xiang Li, Peng Zhu, Yifeng Luo and Yuqi Liang. 'Predicting stock market trends with self-supervised learning'. In: *Neurocomputing* 568 (2024), p. 127033. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2023.127033. URL: https://www.sciencedirect.com/science/article/pii/S0925231223011566.

[258] Jaemin Yoo, Yejun Soun, Yong-chan Park and U Kang. 'Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts'. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 2037–2045. ISBN: 9781450383325. DOI: 10.1145/3447548.3467297. URL: https://doi.org/10.1145/3447548.3467297.

[259] Zinuo You, Pengju Zhang, Jin Zheng and John Cartlidge. 'Multi-Relational Graph Diffusion Neural Network with Parallel Retention for Stock Trends Classification'. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Apr. 2024, pp. 6545–

6549. DOI: 10.1109/icassp48485.2024.10447394. URL: http://dx.doi.org/10.1109/ICASSP48485.2024.10447394.

[260] Xinli Yu, Zheng Chen and Yanbin Lu. 'Harnessing LLMs for Temporal Data - A Study on Explainable Financial Time Series Forecasting'. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Ed. by Mingxuan Wang and Imed Zitouni. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 739–753. DOI: 10.18653/v1/2023.emnlp-industry.69. URL: https://aclanthology.org/2023.emnlp-industry.69.

[261] Ailing Zeng, Muxi Chen, Lei Zhang and Qiang Xu. 'Are transformers effective for time series forecasting?' In: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN: 978-1-57735-880-0. DOI: 10.1609/aaai.v37i9.26317. URL: https://doi.org/10.1609/aaai.v37i9.26317.

[262] Yan Zeng, Xinsong Zhang and Hang Li. 'Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts'. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 25994–26009. URL: https://proceedings.mlr.press/v162/zeng22c.html.

[263] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty and Carsten Eickhoff. 'A Transformer-based Framework for Multivariate Time Series Representation Learning'. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 2114–2124. ISBN: 9781450383325. DOI: 10.1145/3447548.3467401. URL: https://doi.org/10.1145/3447548.3467401.

[264] Donglin Zhan, Yusheng Dai, Yiwei Dong, Jinghai He, Zhenyi Wang and James Anderson. 'Meta-Adaptive Stock Movement Prediction with Two-Stage Representation Learning'. In: *Proceedings of the 2024 SIAM International Conference on Data Mining*, pp. 508–516. DOI: 10.1137/1.9781611978032.59. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611978032.59. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611978032.59.

[265] Jilin Zhang, Lishi Ye and Yongzeng Lai. 'Stock Price Prediction Using CNN-BiLSTM-Attention Model'. In: *Mathematics* 11.9 (2023). ISSN: 2227-7390. DOI: 10.3390/math11091985. URL: https://www.mdpi.com/2227-7390/11/9/1985.

[266] Lan Zhang, Per A. Mykland and Yacine Aït-Sahalia. 'A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data'. In: *Journal of the American Statistical Association* 100.472 (2005), pp. 1394–1411. ISSN: 01621459. URL: http://www.jstor.org/stable/27590680 (visited on 05/05/2024).

[267] Liheng Zhang, Charu Aggarwal and Guo-Jun Qi. 'Stock Price Prediction via Discovering Multi-Frequency Trading Patterns'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 2141–2149. ISBN: 9781450348874. DOI: 10.1145/3097983.3098117. URL: https://doi.org/10.1145/3097983.3098117.

[268] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang and Peide Liu. 'Transformer-based attention network for stock movement prediction'. In: *Expert Systems with Applications* 202 (2022), p. 117239. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.117239. URL: https://www.sciencedirect.com/science/article/pii/S0957417422006170.

[269] Xi Zhang, Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fang and Philip S. Yu. 'Improving stock market prediction via heterogeneous information fusion'. In: *Knowledge-Based Systems* 143 (Mar. 2018), pp. 236–247.

ISSN: 0950-7051. DOI: 10.1016/j.knosys.2017.12.025. URL: http://dx.doi.org/10.1016/j.knosys.2017.12.025.

[270] Zhaofeng Zhang, Banghao Chen, Shengxin Zhu and Nicolas Langrené. *Quantformer: from attention to profit with a quantitative transformer trading strategy*. 2024. arXiv: 2404.00424 [q-fin.MF]. URL: https://arxiv.org/abs/2404.00424.

[271] Feng Zhao, Xinning Li, Yating Gao, Ying Li, Zhiquan Feng and Caiming Zhang. 'Multi-layer features ablation of BERT model and its application in stock trend prediction'. In: *Expert Systems with Applications* 207 (2022), p. 117958. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.117958. URL: https://www.sciencedirect.com/science/article/pii/S0957417422011939.

[272] Pengfei Zhao, Haoren Zhu, Wilfred Siu Hung NG and Dik Lun Lee. 'From GARCH to Neural Network for Volatility Forecast'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.15 (Mar. 2024), pp. 16998–17006. DOI: 10.1609/aaai.v38i15.29643. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29643.

[273] Xiaojun Zhao, Na Zhang, Yali Zhang, Chao Xu and Pengjian Shang. 'Equity markets volatility clustering: A multiscale analysis of intraday and overnight returns'. In: *Journal of Empirical Finance* 77 (2024), p. 101487. ISSN: 0927-5398. DOI: https://doi.org/10.1016/j.jempfin.2024.101487. URL: https://www.sciencedirect.com/science/article/pii/S0927539824000227.

[274] Zhiyong Zhao, Ruonan Rao, Shaoxiong Tu and Jun Shi. 'Time-Weighted LSTM Model with Redefined Labeling for Stock Trend Prediction'. In: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence*. 2017, pp. 1210–1217. DOI: 10.1109/ICTAI.2017.00184.

[275] Xiaolin Zheng, Mengpu Liu and Mengying Zhu. ‘Deep Hashing-based Dynamic Stock Correlation Estimation via Normalizing Flow’. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. Ed. by Edith Elkind. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 4993–5001. DOI: 10.24963/ijcai.2023/555. URL: https://doi.org/10.24963/ijcai.2023/555.

[276] Junxian Zhou, Shoujin Wang and Yuming Ou. ‘Fourier Graph Convolution Transformer for Financial Multivariate Time Series Forecasting’. In: *2024 International Joint Conference on Neural Networks*. 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650090.

[277] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun and Rong Jin. ‘FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting’. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 27268–27286. URL: https://proceedings.mlr.press/v162/zhou22g.html.

[278] Tian Zhou, Peisong Niu, xue wang xue, Liang Sun and Rong Jin. ‘One Fits All: Power General Time Series Analysis by Pretrained LM’. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 43322–43355. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/86c17de05579cde52025f9984e6e2ebb-Paper-Conference.pdf.

[279] Zhihan Zhou, Liqian Ma and Han Liu. ‘Trade the Event: Corporate Events Detection for News-Based Event-Driven Trading’. In: *Findings of the Association for Computational Linguistics: 2021 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*. Ed. by Chengqing Zong, Fei Xia, Wenjie

Li and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 2114–2124. DOI: 10.18653/v1/2021.findings-acl.186. URL: https://aclanthology.org/2021.findings-acl.186.

[280] Jianping Zhu, Xin Guo, Yang Chen, Yao Yang, Wenbo Li, Bo Jin and Fei Wu. 'Adaptive Meta-Learning Probabilistic Inference Framework for Long Sequence Prediction'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.15 (Mar. 2024), pp. 17159–17166. DOI: 10.1609/aaai.v38i15.29661. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29661.

[281] Jinan Zou, Haiyao Cao, Lingqiao Liu, Yuhao Lin, Ehsan Abbasnejad and Javen Qinfeng Shi. 'Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model'. In: *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing*. Ed. by Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura and Hsin-Hsi Chen. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 178–186. DOI: 10.18653/v1/2022.finnlp-1.24. URL: https://aclanthology.org/2022.finnlp-1.24.

[282] Jinan Zou, Qingying Zhao, Yang Jiao, Haiyao Cao, Yanxi Liu, Qingsen Yan, Ehsan Abbasnejad, Lingqiao Liu and Javen Qinfeng Shi. *Stock Market Prediction via Deep Learning Techniques: A Survey*. 2023. arXiv: 2212.12717 [q-fin.GN].

[283] IEEE. *IEEE Standard for Transparency of Autonomous Systems*. IEEE Std 7001-2021, 2021. URL: https://standards.ieee.org/ieee/7001/10282/.

[284] IEEE. *IEEE Standard for Algorithmic Bias Considerations*. IEEE Std 7003-2024, 2024. URL: https://standards.ieee.org/ieee/7003/11285/.

[285] Renuka Sharma and Kiran Mehta. *Deep Learning Tools for Predicting Stock Market Movements*. John Wiley & Sons, 2024. URL: https://books.google.de/books?id=2joCEQAAQBAJ.

[286] N. Srinivasan. *Stock Price Prediction: A Referential Approach on How to Predict the Stock Price Using Simple Time Series*. Clever Fox Publishing, 2023. URL: https://books.google.de/books?id=Aj8jEAAAQBAJ.

[287] Aakanksha Jadhav and Vishal Mirza. *Large Language Models in equity markets: applications, techniques, and insights*. Frontiers in Artificial Intelligence, Volume 8, 2025. DOI: https://doi.org/10.3389/frai.2025.1608365. URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1608365.

[288] Ruonan Wu and Hong Liu. *A survey on the application and research progress of large language models in financial forecasting*. AIP Advances, 15(6):060704, 2025. DOI: https://doi.org/10.1063/5.0274031.

[289] Bingxing Wang. *Empirical Evaluation of Large Language Models for Asset-Return Prediction*. Academic Journal of Sociology and Management, Vol. 3(4), pp. 18–25, July 2025. DOI: 10.70393/616a736d.333035. URL: https://www.suaspress.org/ojs/index.php/AJSM/article/view/v3n4a03.

[290] Rico Sennrich, Barry Haddow, and Alexandra Birch. 'Neural Machine Translation of Rare Words with Subword Units'. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. doi: 10.18653/v1/P16-1162. url: https://aclanthology.org/P16-1162/.

[291] Taku Kudo and John Richardson. 'SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text

Processing'. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2018. doi: 10.18653/v1/D18-2012. url: https://arxiv.org/abs/1808.06226.

[292] Jianlin Su, Yu Lu, Shengfeng Pan, et al. 'RoFormer: Enhanced Transformer with Rotary Position Embedding'. 2021. arXiv: 2104.09864. doi: 10.48550/arXiv.2104.09864. url: https://arxiv.org/abs/2104.09864.

[293] Ofir Press, Noah A. Smith, and Mike Lewis. 'Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation'. 2021. arXiv: 2108.12409. doi: 10.48550/arXiv.2108.12409. url: https://arxiv.org/abs/2108.12409.

[294] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, et al. 'Rethinking Attention with Performers'. In: International Conference on Learning Representations. 2021. arXiv: 2009.14794. doi: 10.48550/arXiv.2009.14794. url: https://arxiv.org/abs/2009.14794.

[295] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 'FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness'. In: Advances in Neural Information Processing Systems. 2022. doi: 10.48550/arXiv.2205.14135. url: https://arxiv.org/abs/2205.14135.

[296] Mike Lewis, Yinhan Liu, Naman Goyal, et al. 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension'. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. arXiv: 1910.13461. doi: 10.48550/arXiv.1910.13461. url: https://arxiv.org/abs/1910.13461.

[297] Zhilin Yang, Zihang Dai, Yiming Yang, et al. 'XLNet: Generalized Autoregressive Pretraining for Language Understanding'. In: Advances in Neural Information Processing Systems. 2019. arXiv: 1906.08237. doi:

10.48550/arXiv.1906.08237. url: https://arxiv.org/abs/1906.08237.

[298] Yinhan Liu, Myle Ott, Naman Goyal, et al. 'RoBERTa: A Robustly Optimized BERT Pretraining Approach'. 2019. arXiv: 1907.11692. doi: 10.48550/arXiv.1907.11692. url: https://arxiv.org/abs/1907.11692.

[299] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. 'Parameter-Efficient Transfer Learning for NLP'. 2019. arXiv: 1902.00751. doi: 10.48550/arXiv.1902.00751. url: https://arxiv.org/abs/1902.00751.

[300] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. 'LoRA: Low-Rank Adaptation of Large Language Models'. 2022. arXiv: 2106.09685. doi: 10.48550/arXiv.2106.09685. url: https://arxiv.org/abs/2106.09685.

[301] Xiang Lisa Li and Percy Liang. 'Prefix-Tuning: Optimizing Continuous Prompts for Generation'. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. 2021. doi: 10.18653/v1/2021.acl-long.353. url: https://aclanthology.org/2021.acl-long.353/.

[302] Brian Lester, Rami Al-Rfou, and Noah Constant. 'The Power of Scale for Parameter-Efficient Prompt Tuning'. 2021. arXiv: 2104.08691. doi: 10.48550/arXiv.2104.08691. url: https://arxiv.org/abs/2104.08691.

[303] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. 'Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks'. In: Advances in Neural Information Processing Systems. 2020. arXiv: 2005.11401. doi: 10.48550/arXiv.2005.11401. url: https://arxiv.org/abs/2005.11401.

[304] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, et al. 'Dense Passage Retrieval for Open-Domain Question Answering'. In: Proceedings of

the 2020 Conference on Empirical Methods in Natural Language Processing. 2020. arXiv: 2004.04906. doi: 10.48550/arXiv.2004.04906. url: https://arxiv.org/abs/2004.04906.

[305] Andrew Trask, Felix Hill, Scott Reed, Jack Rae, Chris Dyer, and Phil Blunsom. 'Neural Arithmetic Logic Units'. 2018. arXiv: 1808.00508. doi: 10.48550/arXiv.1808.00508. url: https://arxiv.org/abs/1808.00508.

[306] Shijie Wu, Ozan Irsoy, Steven Lu, et al. 'BloombergGPT: A Large Language Model for Finance'. 2023. arXiv: 2303.17564. doi: 10.48550/arXiv.2303.17564. url: https://arxiv.org/abs/2303.17564.

[307] Dogu Araci. 'FinBERT: Financial Sentiment Analysis with Pretrained Language Models'. 2019. arXiv: 1908.10063. doi: 10.48550/arXiv.1908.10063. url: https://arxiv.org/abs/1908.10063.

[308] Edwin J. Elton, Martin J. Gruber, Stephen J. Brown, and William N. Goetzmann. *Modern Portfolio Theory and Investment Analysis*. 2009. 7th ed. Wiley. ISBN: 9780470050828. url: https://books.google.com/books/about/Modern_Portfolio_Theory_and_Investment_A.html?id=aOtcTEQ3DAUC.

[309] William F. Sharpe. "The Sharpe Ratio". 1994. *The Journal of Portfolio Management* 21(1): 49–58. doi: 10.3905/jpm.1994.409501. url: https://www.pm-research.com/content/iijpormgmt/21/1/49.

[310] Richard C. Grinold and Ronald N. Kahn. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk*. 2000. 2nd ed. McGraw–Hill. ISBN: 9780070248823.

[311] Alexei Chekhlov, Stanislav Uryasev, and Michael Zabarankin. "Drawdown Measure in Portfolio Optimization". 2005. *International Journal of Theoretical and Applied Finance* 8(1): 13–58. doi:

10.1142/S0219024905002767. url: https://www.worldscientific.com/doi/10.1142/S0219024905002767.

[312] Richard A. Brealey, Stewart C. Myers, and Franklin Allen. *Principles of Corporate Finance*. 2020. 13th ed. McGraw–Hill Education. ISBN: 9781260013900. url: https://www.valore.com/products/principles-of-corporate-finance/9781260013900.

[313] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008. Cambridge University Press. ISBN: 9780521865715. url: https://nlp.stanford.edu/IR-book/

[314] Stanley F. Chen and Joshua Goodman. "An Empirical Study of Smoothing Techniques for Language Modeling." 1999. Harvard University, Technical Report TR-10-98 (revised). url: https://www.microsoft.com/en-us/research/publication/an-empirical-study-of-smoothing-techniques-for-language-modelin

[315] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." 1989. *Proceedings of the IEEE*, 77(2):257–286. url: https://ieeexplore.ieee.org/document/18626

[316] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." 2001. In: *Proceedings of ICML*. url: https://dl.acm.org/doi/10.5555/645530.655813

[317] Joakim Nivre. "Algorithms for Deterministic Incremental Dependency Parsing." 2008. *Computational Linguistics*, 34(4):513–553. url: https://direct.mit.edu/coli/article/34/4/513/1581/Algorithms-for-Deterministic-Incremental

[318] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 2009. 2nd ed. Prentice Hall. ISBN: 9780131873216. url: https://web.stanford.edu/~jurafsky/slp3/

[319] Fabrizio Sebastiani. “Machine Learning in Automated Text Categorization.” 2002. *ACM Computing Surveys*, 34(1):1–47. url: https://dl.acm.org/doi/10.1145/505282.505283

Appendix A

A.1 Normalization Module

For SMP, primary reliance is placed on the normalization module introduced in [168], which has been reimplemented based on the information provided in the original publication. This reimplementation was substantially modified to improve performance within the models and to enhance numerical stability. The adapted version of the approach is (using the original notation from [168])

$$\begin{aligned} \beta(x)' &= \mathbf{W}_\beta s_\beta + \mathbf{b}_\beta \\ \beta(x)'' &= \beta(x)' + \dot{\mathbf{b}}_\beta \\ \beta(x)''' &= \beta(x)''^{-1} \\ \beta(x) &= f_{ln}(\beta(x)''') \end{aligned}$$

with $\dot{\mathbf{b}} \in \{\epsilon, -\epsilon\}^\xi$ and

$$\forall i : \beta(x)'[i] > 0 : \dot{\mathbf{b}}_\beta[i] = \epsilon_{\text{norm}} \oplus \forall \beta(x)'[j] \leq 0 : \dot{b}_\beta[j] = -\epsilon_{\text{norm}} \tag{A.1}$$

and

$$x' = \sigma((x - \alpha(x)) \odot \beta(x)) \tag{A.2}$$

and

$$\mathbf{x}'' = f_{\text{ln}}(\tanh(((\mathbf{x} - \boldsymbol{\alpha}(x)) \odot \boldsymbol{\beta}(x)) \odot \gamma(x) \cdot \alpha_{\text{norm}})) \; . \tag{A.3}$$

A.2 Market data

Since complete market or index data are generally not publicly available, price trends from ETFs representing the respective markets have been used instead. The ETFs employed for the different countries are Amundi MSCI France UCITS ETF 🇫🇷, iShares MSCI Canada ETF 🇨🇦, iShares Core DAX UCITS ETF 🇩🇪, iShares MSCI China UCITS ETF 🇨🇳, iShares MSCI India ETF 🇮🇳, iShares MSCI Japan UCITS ETF 🇯🇵, SPDR S&P 500 ETF 🇺🇸, Vanguard Total Stock Market ETF 🇺🇸, Vanguard FTSE 100 UCITS ETF 🇬🇧, CAC PAR 🇫🇷, China PAR 🇨🇳, DAX 🇩🇪, DOW Jones 🇺🇸, FTFX 🇫🇷, INDX.SAO 🇧🇷, MXE 🇲🇽, RYJSX 🇯🇵, Sensexbees 🇮🇳, Zag TRT 🇨🇦. The selection of countries was guided by their gross national product; however, it is significantly constrained by data availability, as several major economies (such as South Korea, Russia or Indonesia) are not represented.

A.3 Stock Splits

A stock split is when a company increases the number of its stocks by dividing existing stocks into multiple ones, typically to make the stock more affordable and attractive to investors while maintaining the same overall market value[1]. Vice versa a reverse stock split is when a company reduces the number of its outstanding stocks by consolidating them into fewer, higher-priced stocks, typically to increase the stock price and maintain exchange listing requirements or improve its perception in the market. A straightforward approach is adopted, whereby a threshold value is selected as θ_{Split} and if $\overline{\underline{\left(\frac{x_i^{(t)}[4]}{x_i^{(t+1)}[4]}\right)}} \geq \theta_{\text{Split}}$ applies, $\alpha_{\text{Split}} = \frac{\mathbf{x}_i^{(t)}[4]}{\mathbf{x}_i^{(t+1)}[4]}$ and

$$\forall j : t \leq j \leq \Delta t, \forall f : 1 \leq f \leq \mathbb{F} : x_i^{(j)} \leftarrow x_i^{(j)}[f] \cdot \alpha_{\text{Split}}[f] \tag{A.4}$$

are defined.

[1] https://www.finra.org/investors/investing/investment-products/stocks/stock-splits

Since relative returns are used, this is only necessary if the split occurs within X, but not for the entire $\grave{X}$. To evaluate the effectiveness of the approach is was applied to selected stock splits from the S&P-500 🇺🇸, CSI-300 🇨🇳, and DAX-40 🇩🇪. Splits are identified through https://finance.yahoo.com/calendar/splits/. The approach yielded satisfactory results across these indices.

A.4 SPP Implementation

The non-stationarity of the stock time series poses particular challenges for the selection of the SPP target variable. Through empirical analysis, it has been found that predicting relative returns yields the best results for SPP, since the absolute returns, RLR, max-min normalized prices or the like suffer too much from non-stationarity. Since these are small decimal fractions, $\mathcal{L}$ can become too small to effectively calculate the weights. A simple yet effective trick is applied in the experiments, wherein multiplication by the scaling factor $\alpha_{\text{SPP}} \gg 1$ is performed. Therefore $\breve{X} \in \mathbb{R}^{|C| \times \Delta t \times \mathbb{F}}$ is defined as the raw OHCLV features without any normalization. Further the targets for training step $j \sim \mathcal{U}(\mathbb{N} < \mathbb{T})$ are defined as

$$\mathbf{y}[i] = \frac{\breve{X}[i, j+\omega, 4] - \breve{X}[i, j, 4]}{\breve{X}[i, j, 4]} \cdot \alpha_{\text{SPP}} \quad \forall i \in \mathbb{N} < |C| \; . \tag{A.5}$$

Since no activation function is applied after the final layer in SPP, arbitrary values can be predicted and subsequently converted back to prices relative to the original input. To recover the original target from the relative return or the absolute predicted price; $\frac{\hat{y}}{\alpha_{\text{SPP}}} \cdot x_i^j + x_i^j$ can be used.

A.5 SMP Prediction Distribution

With SMP, the phenomenon can occur that the same movement is always predicted for many c_i across all batches and training instances. The problem, especially for weaker performing baseline models is that the accuracy achieved cannot necessarily be surpassed. Although this represents a valid solution, such trivial predictions are intentionally avoided. Therefore, only predictions whose distribution differs from that of the solution in the validation or test data by no more than ϵ are included in the evaluation.

Formally, if

$$\forall c_i : \left| \left(\frac{1}{|I|} \sum_{j \in I} \mathbf{y}_i^{(j)} \right) - \left(\frac{1}{|I|} \sum_{j \in I} \hat{\mathbf{y}}_i^{(j)} \right) \right| < \epsilon \tag{A.6}$$

applies with I being the set of all predictions in the run, it is considered valid run to compare the performance to the others. The distribution is approximated using a 50/50 label split across all datasets, enabling the evaluation to proceed without requiring prior knowledge of the label distribution in the test and validation sets. To overcome the local optimum in Θ during training, the loss penalty

$$\mathcal{L}_p = \frac{1}{|C|} \sum_{c_i \in C} ((\sum_{b=0}^{\beta} \tanh(\lambda_{\text{SMP}} \cdot \hat{Y}_i^{(b)}))^2 + \epsilon_p)^{0.5} \tag{A.7}$$

is applied (with β as the mini batch size) inspired by the hinge loss in order to force the model to produce weighted/balanced predictions for each stock. The whole loss is expressed as $\mathcal{L}_{\text{SMP}} \leftarrow \mathcal{L}_{\text{SMP}} + \lambda_p \cdot \mathcal{L}_p$.

A.6 Data Cleaning

Despite all efforts, the problem of missing values becomes more severe as the time resolution increases—particularly for intraday data, where entire time slices may be missing. As previously discussed, although the choice between forward filling, backward filling, and linear interpolation remains a subject of debate, the latter approach is adopted in this work. Linear interpolation is employed, as backward filling was found to simplify the SPP task excessively, resulting in unrealistically high performance. However, time steps where too much data is missing are excluded to ensure that interpolation remains meaningful. To address this issue, a relative threshold θ_{missing} is defined: if more than θ_{missing} percent of stocks lack data at a given time step (e.g., a specific day or time slice), that time step is filtered out and excluded from use. As this problem predominantly affects older intraday data from the early 2000s, its impact on the overall evaluation remains limited.

A.7 Review System

As in [282], the search is conducted using Google Scholar, focusing on the conferences ACL, EMNLP, AAAI, IJCAI, ICAIF, NeurIPS, and KDD, with the following keywords: ‘stock prediction, market, finance and portfolio’, as well as the additional NLP-adaption specific keywords (see Section 2.3) and also include the keywords ‘RNN, LSTM, GNN and Transformer’. In contrast to [282], papers with fewer than two pages are not excluded. The keyword ‘RL’ is omitted, as it is not relevant to the scope of the present thesis. However, the keyword ‘portfolio’ is retained, as this PhD thesis—although focused on SF—benefits from its inclusion: on the one hand, portfolio optimization encompasses numerous approaches relevant to W2V adaptations, Doc2Vec adaptations, and contextualized embeddings; on the other hand, many optimization papers refer to methods pertinent to SMP/SPP tasks. In addition, non-relevant papers were manually filtered to replace for the automated process of ‘using machine learning to filter papers that predicted the stock market’ [282]. All papers identified through this method were screened, as in [282], and those deemed relevant are included in Chapter 2.

A.8 Reverse Function Definition

Reversing the digits is defined as

$$
\begin{aligned}
f_{\text{reverse}}(z) &= (z_l, z_{l-1}, \dots, z_1)\ , \\
z_i &= \left\lfloor \frac{z - v_i}{10^{l-i}} \right\rfloor \quad \text{and} \\
v_i &= \begin{cases} 0 & \text{if } i = 0 \\ \left\lfloor \frac{z}{10^{l-i+1}} \right\rfloor \cdot 10^{l-i+1} & \text{else} \end{cases}
\end{aligned}
$$

, where $l = \lfloor \log_{10} x + 1 \rfloor$ holds.

A.9 Relevance Visualization

A modified relevance visualization, previously presented in [221] and [222] and originally inspired by [17], is employed. For the masking tasks

$$R = \bar{A} \text{ and } \nabla A := \frac{\partial \frac{1}{\mathbf{1}^T \ddot{M} \mathbf{1}} \cdot \mathbf{1}(F^{\langle \mathrm{F} \rangle}(X) \otimes M \otimes \ddot{M})\mathbf{1}^T}{\partial A} \tag{A.8}$$

with

$$\begin{aligned} \ddot{M} \in \{0,1\}^{\dim(X)} : \forall i,j : \ddot{M}[i,j] = 1 \Longrightarrow \\ (F^{\langle \mathrm{F} \rangle}(X)[i,j] = 0 \wedge X[i,j] \leq 0) \oplus \ (F^{\langle \mathrm{F} \rangle}(X)[i,j] = 1 \wedge X[i,j] > 0) \end{aligned}$$

is calculated following the notation of [17].