

BACHELOR THESIS
Gianluca Löwe

KI-basiertes Assistenzsystem für ältere Menschen mit Hilfe einer Smartwatch

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Gianluca Löwe

KI-basiertes Assistenzsystem für ältere Menschen mit Hilfe einer Smartwatch

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Wirtschaftsinformatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Kai von Luck
Zweitgutachter: Prof. Susanne Draheim

Eingereicht am: 22.01.2026

Gianluca Löwe

Thema der Arbeit

KI-basiertes Assistenzsystem für ältere Menschen mit Hilfe einer Smartwatch

Stichworte

Ambient-Assisted-Living, Assistenzsysteme, Smartwatch, Wearables, MS-TCN, Nudging, Trinkgestenerkennung, Companions, Quantified Self

Kurzzusammenfassung

Der demografische Wandel führt zu einer zunehmenden Alterung der Gesellschaft und einem damit verbundenen Anstieg des Pflegebedarfs. Ambient Assisted Living (AAL) Ansätze zielen darauf ab, ältere Menschen durch technische Assistenzsysteme bei alltäglichen Aufgaben zu unterstützen, ihre Selbstständigkeit zu fördern und Aufgabenbereiche der ambulanten Pflege zu entlasten. In dieser Arbeit wird ein Assistenzsystem vorgestellt, das auf einer handelsüblichen Smartwatch basiert und den Schwerpunkt auf eine Trinkgestenerkennung mithilfe eines Multi-Stage Temporal Convolutional Network (MS-TCN) setzt. Außerdem wird eine Bewegungsanalyse anhand von Herzfrequenzen, Schrittzahlen und GPS-Daten durchführt. Ziel ist es, ältere Menschen daran zu erinnern, regelmäßig zu trinken und sich ausreichend zu bewegen, um Dehydrierung, Einschränkungen der Mobilität und damit verbundene Gesundheitsrisiken zu vermeiden. Es werden die Entwicklung und Implementierung des Modells beschrieben. Die Ergebnisse zeigen, dass die Sensoren einer Smartwatch ausreichend Genauigkeit bieten um im Zusammenspiel mit einem MS-TCN-Modell Trinkgesten zuverlässig zu erkennen und somit als Grundlage für ein effektives Assistenzsystem dienen können.

Gianluca Löwe

Title of Thesis

AI-based assistance system for older people using a smartwatch

Keywords

Ambient Assisted Living, Assistive Systems, Smartwatch, Wearables, MS-TCN, Nudging, Drinking Gesture Detection, Companions, Quantified Self

Abstract

Demographic change is leading to an aging society and a corresponding increase in the need for care. Ambient Assisted Living (AAL) approaches aim to support older people in their everyday tasks with technical assistance systems, promote their independence, and relieve the burden on outpatient care providers. This paper presents an assistance system based on a commercially available smartwatch that focuses on recognizing drinking gestures using a multi-stage temporal convolutional network (MS-TCN). In addition, motion analysis is performed based on heart rates, step counts, and GPS data. The aim is to remind older people to drink regularly and get enough exercise to avoid dehydration, mobility restrictions, and associated health risks. The development and implementation of the model are described. The results show that the sensors of a smartwatch offer sufficient accuracy to reliably recognize drinking gestures in conjunction with an MS-TCN model and can therefore serve as the basis for an effective assistance system.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
1 Einleitung	1
1.1 Motivation	1
1.2 Aufbau der Arbeit	2
2 Theoretische Grundlagen	3
2.1 (De-)Hydration im Alter	3
2.2 Bewegung im Alter	4
2.3 Smartwatches als Companions	4
2.3.1 Sensorik	5
2.4 Ambient Assisted Living	5
2.5 Quantified Self	6
2.6 Nudging	6
2.7 Verwandte Gesten	6
2.8 Verwandte Arbeiten	7
2.8.1 Trinkgestenerkennung mit am Handgelenk getragener IMU	7
2.8.2 Becherbasierte Systeme mit Lastsensor oder integriertem IMU	8
3 Design und Implementierung	10
3.1 PC-Spezifikationen, Tools und Projektumgebungen	11
3.2 Anforderungen	11
3.3 Trainingsmodus	13
3.3.1 In dieser Arbeit verwendete Trainingsdaten	13
3.3.2 Evaluationsdaten	14
3.4 Systemarchitektur	14
3.4.1 Datenbank	15

3.4.2	Smartwatch-Applikation	15
3.4.3	Server	19
3.5	Implementierungsdetails	22
3.5.1	Ablauf der Analyse der Tagesdaten auf Trinkgesten	22
3.5.2	Ablauf der Analyse der Tagesdaten auf Bewegung	24
3.6	Datenübertragung	26
3.7	Zusatzfunktionen	27
4	Ergebnisse	28
4.1	Modellwahl	28
4.2	Trainingssession-Versuche	33
4.2.1	Finetuning	34
4.3	Einschränkungen	34
4.3.1	Akkulaufzeit	35
5	Fazit und Ausblick	37
	Literaturverzeichnis	39
	Selbstständigkeitserklärung	42

Abbildungsverzeichnis

3.1	Übersicht der Systemkomponenten	10
3.2	Einrichtungsschritte der Smartwatch-Applikation	16
3.3	Kennzeichnen einer Trinkgeste während einer Trainingssession	18
3.4	Ablaufdiagramm zur Analyse der Tagesdaten auf Trinkgesten	23
3.5	Beispiel Push-Benachrichtigung für eine Trinkerinnerung	24
3.6	Ablaufdiagramm zur Analyse der Tagesdaten auf Bewegung	25
3.7	Beispiel Push-Benachrichtigung für eine Bewegungserinnerung	26
3.8	Manuelle Abfrage des Hydrations- und Bewegungsstatus	27
4.1	Durchschnittliche absolute Abweichung aller Modelle trainiert mit ver- schiedener Segmentierung der Trainingsdaten.	29
4.2	Durchschnittliche absolute Abweichungen der besten Kombinationen aus Abbildung 4.1 auf den True Positives.	31

Tabellenverzeichnis

3.1	Funktionale Anforderungen	12
3.2	Nicht-funktionale Anforderungen	12
4.1	Durchschnittliche Abweichungen der besten Modelle	32
4.2	Durchschnittliche segmentale Metriken der besten Kombinationen aus Sequenzlängen und Überlappungsraten über 10 Trainingsläufe	32
4.3	Durchschnittliche frame-wise Metriken der besten Kombinationen aus Sequenzlängen und Überlappungsraten über 10 Trainingsläufe	32
4.4	Weitere frame-wise Metriken	33

1 Einleitung

1.1 Motivation

Der demografische Wandel in Deutschland stellt die Gesellschaft vor zunehmende Herausforderungen. Besonders der Pflegebereich ist durch den steigenden Anteil der Pflegebedürftigen erheblich belastet [11]. So ist die Anzahl der Pflegebedürftigen im Zeitraum von 2013 bis 2023 von 2,63 Millionen Menschen auf 5,69 Millionen gestiegen, [17] was einem Anstieg von 119% entspricht, wohingegen sich die Anzahl der Pflegekräfte in Pflegeheimen und ambulanten Pflegediensten im selben Zeitraum lediglich von rund 1 Million auf 1,2 Millionen erhöht hat [16]. Prognosen deuten darauf hin, dass sich der Fachkräftemangel in der Pflege in den kommenden Jahrzehnten weiter verschärfen wird [11].

Vor diesem Hintergrund gewinnt der Ambient Assisted Living (AAL) Ansatz an Bedeutung. Der Begriff beschreibt den Einsatz von technischen Assistenzsystemen, um ältere Menschen bei alltäglichen Aufgaben zu unterstützen [1]. Manche dieser AAL-Technologien bieten außerdem eine Möglichkeit, alltägliche Aktivitäten unauffällig zu erfassen und auszuwerten, um daraus Rückschlüsse auf den Gesundheitszustand der Nutzer zu ziehen. Entsprechend dem Quantified-Self-Ansatz werden Alltagsdaten durch verschiedene Sensorik aufgenommen, analysiert und die Ergebnisse dem Nutzer im Anschluss mitgeteilt. Insbesondere ältere Menschen vergessen häufig, regelmäßig zu trinken oder sich ausreichend zu bewegen [2]. Ursache hierfür ist unter anderem ein im Alter vermindertes Durstempfinden [7]. Auf der Basis von erfassten Sensordaten können Assistenzsysteme ältere Menschen auf minimal-invasive Weise dabei unterstützen, regelmäßig genügend Flüssigkeit zu sich zu nehmen und ihre tägliche Bewegung zu erhöhen.

Das in dieser Arbeit vorgestellte Assistenzsystem unterstützt die ambulante Pflege in genau diesen beiden Aufgabenfeldern. Es umfasst sowohl die Erkennung von Trinkges-ten mithilfe eines Multi-Stage Temporal Convolutional Networks (MS-TCN) als auch eine Bewegungsanalyse anhand von Herzfrequenzen, Schrittzahlen und GPS-Daten. Das

System begleitet den Nutzer unauffällig im Alltag, sammelt relevante Daten und weist bei der Unterschreitung definierter Mindestwerte darauf hin, mehr zu trinken oder sich mehr zu bewegen. Die einzigen Voraussetzungen sind das Tragen sowie die einmalige Einrichtung der Smartwatch.

Ziel dieser Arbeit ist es, zu untersuchen, inwieweit Smartwatches geeignet sind, Trinkgesten zuverlässig und konsistent zu erkennen und Nutzer nachhaltig dazu zu motivieren, eine ausreichende Flüssigkeitszufuhr, sowie ein angemessenes Maß an körperlicher Aktivität einzuhalten.

1.2 Aufbau der Arbeit

Diese Arbeit ist in fünf Kapitel unterteilt. Im Anschluss an die Einleitung werden im zweiten Kapitel die theoretischen Grundlagen inklusive dem Stand der Forschung, das Problem der Trinkgestenerkennung und der Bewegungsanalyse vorgestellt. Das darauffolgende Kapitel erläutert die technischen Details der Implementierung. Danach werden die Ergebnisse der Arbeit diskutiert und im letzten Kapitel werden Fazit und Ausblick diskutiert.

2 Theoretische Grundlagen

2.1 (De-)Hydration im Alter

Mit zunehmendem Alter nimmt das Durstempfinden ab, wodurch ältere Menschen häufig weniger trinken, selbst wenn bereits ein Flüssigkeitsdefizit vorliegt. Eine anhaltende oder wiederholte Dehydration kann im Alter erhebliche gesundheitliche Folgen haben. Dazu zählen unter anderem Konzentrationsstörungen, Verwirrtheit, Schwindel und eine erhöhte Sturzgefahr. In schweren Fällen führt Dehydration zu vermehrten Krankenhausaufenthalten [6, 14].

Vor diesem Hintergrund ist eine gleichmäßige und über den Tag verteilte Flüssigkeitsaufnahme von besonderer Bedeutung. Eine kontinuierliche Hydration unterstützt nicht nur die Vitalfunktionen des Körpers, sondern trägt auch zur Erhaltung der kognitiven Leistungsfähigkeit bei [10].

Assistive Systeme, die das Trinkverhalten automatisiert erfassen und auswerten, können hier einen wichtigen Beitrag leisten. Durch die kontinuierliche Auswertung von gesammelten Tagesdaten und der Ableitung des Hydrationsstatus lassen sich potenzielle Defizite frühzeitig erkennen und durch gezielte Benachrichtigungen präventiv entgegenwirken.

Laut der European Society for Clinical Nutrition and Metabolism (ESPEN) soll ein erwachsener Mann über 65 Jahre etwa 2,5 L und eine erwachsene Frau etwa 2 L Flüssigkeit pro Tag zu sich nehmen. Dies beinhaltet die Flüssigkeitsaufnahme durch Essen. Davon sollen für Männer etwa 2 L und für Frauen etwa 1,6 L durch Getränke hinzugeführt werden [20].

2.2 Bewegung im Alter

Regelmäßige körperliche Aktivität spielt im höheren Lebensalter eine zentrale Rolle für den Erhalt der Gesundheit und der Selbstständigkeit. Sie trägt wesentlich zur Stabilisierung des Herz-Kreislauf-Systems bei und wirkt sich positiv auf die kognitive Leistungsfähigkeit, das psychische Wohlbefinden sowie auf die Prävention chronischer Erkrankungen aus [2].

Die Weltgesundheitsorganisation (WHO) empfiehlt für ältere Erwachsene eine wöchentliche Dauer von 150 bis 300 Minuten moderater aerober körperlicher Aktivität. Als moderate aerobe Aktivität gelten beispielsweise zügiges Gehen mit einer Geschwindigkeit von über 5 km/h oder Radfahren. Wichtig hierbei ist, dass sich die Herzfrequenz merklich für eine längere Zeit erhöht [2].

Trotz der bekannten gesundheitlichen Vorteile bewegen sich viele ältere Menschen im Alltag zu wenig. Neben gezielten Trainingseinheiten ist auch die Alltagsbewegung von großer Bedeutung. Bereits regelmäßiges Gehen, Schwimmen oder Fahrradfahren können einen positiven Effekt auf die Gesundheit haben und damit die Selbstständigkeit fördern [2].

Assistive Systeme, die Bewegungsdaten kontinuierlich und automatisiert erfassen und auswerten, bieten die Möglichkeit, ältere Menschen bei einem aktiveren Lebensstil zu unterstützen. Durch die Kombination von Schrittzähler, GPS und Herzfrequenzsensoren kann das Bewegungsverhalten analysiert werden und auf dessen Grundlage gezielte Benachrichtigungen zur Bewegungsförderung ausgelöst werden, um langfristig die Gesundheit und Mobilität im Alter zu erhalten.

2.3 Smartwatches als Companions

Companions bezeichnen Systeme, die Nutzer im Alltag begleiten und sie unterstützen, erinnern oder motivieren.

Smartwatches zählen zur Kategorie der Wearables und sind am Körper tragbare Systeme, die kontinuierlich Daten über den Nutzer erfassen können. Durch ihre körpernahe Platzierung am Handgelenk eignen sie sich besonders gut für die alltagsnahe Erhebung relevanter Bewegungsdaten. Im Gesundheitskontext werden Smartwatches zunehmend zur Überwachung von Vitalfunktionen eingesetzt [12].

Sie bieten den Vorteil, dass Messungen in realen Alltagssituationen erfolgen und nicht auf künstliche Testumgebungen beschränkt sind. Insbesondere für ältere Menschen stellen Smartwatches eine niedrighschwellige Technologie dar, da sie unauffällig getragen werden können und auch ohne Interaktion funktionieren. So können sie als persönlicher digitaler Companion fungieren, der im Hintergrund agiert und nur bei Bedarf aktiv in Erscheinung tritt.

2.3.1 Sensorik

Moderne Smartwatches verfügen über eine Vielzahl integrierter Sensoren. Zu den zentralen Bewegungssensoren zählen der Beschleunigungsmesser und der Gyroskopsensor. Zusammen bilden sie eine Inertiale Messeinheit (IMU). Der Beschleunigungsmesser misst lineare Beschleunigungen entlang mehrerer Achsen und ermöglicht die Erkennung von Aktivitätsmustern. Der Gyroskopsensor erfasst Rotationsbewegungen und ergänzt die Beschleunigungsdaten um Informationen zur Orientierung und Winkelgeschwindigkeit. Die Kombination beider Sensoren bildet die Grundlage für eine detaillierte Analyse komplexer Bewegungsabläufe, wie sie beispielsweise bei Trinkgesten auftreten [22].

Für die Analyse der körperlichen Aktivität werden zusätzlich physiologische und positionsbezogene Sensoren genutzt. Der Herzfrequenzsensor ermöglicht die Erfassung der momentanen Herzfrequenz und liefert damit Hinweise auf die Intensität körperlicher Aktivitäten. GPS-Sensoren erlauben die Bestimmung von Position, Geschwindigkeit und zurückgelegter Distanz und sind insbesondere für die Bewertung von Aktivitäten im Außenbereich relevant.

2.4 Ambient Assisted Living

Ambient Assisted Living (AAL) beschreibt den Einsatz von technischen Assistenzsystemen, die älteren Menschen dabei unterstützen, ein selbstbestimmtes und unabhängiges Leben in ihrem gewohnten Umfeld zu führen und ihre Lebensqualität zu verbessern. Diese Assistenzsysteme unterstützen ältere Menschen meist in den Bereichen Gesundheit, Sicherheit und der häuslichen Pflege. Dabei werden intelligente Systeme eingesetzt und in den Alltag der Nutzer integriert. Sie können beispielsweise Stürze erkennen, an die Einnahme von Medikamenten erinnern oder die Vitalfunktionen überwachen [1].

2.5 Quantified Self

Quantified Self (QS) bezeichnet die Praxis der systematischen Erfassung und Analyse persönlicher Daten, um Erkenntnisse über das eigene Verhalten und die Gesundheit zu gewinnen [4]. Durch den Einsatz von Wearables wie Smartwatches können Daten kontinuierlich zu verschiedenen Aspekten des Lebens gesammelt werden, darunter körperliche Aktivität, Schlafmuster und Herzfrequenzen. Die gewonnenen Daten werden genutzt, um Muster zu erkennen, Fortschritte zu verfolgen und informierte Entscheidungen zur Verbesserung der Gesundheit und des Wohlbefindens zu treffen.

2.6 Nudging

Der Begriff „Nudging“ beschreibt gezielte, meist subtile Anstöße, die Menschen dabei unterstützen sollen, gesundheitsförderliche Entscheidungen zu treffen, ohne ihre Entscheidungsfreiheit einzuschränken. Im Gegensatz zu verpflichtenden Maßnahmen basiert Nudging auf Freiwilligkeit. Die betroffene Person kann sich jederzeit gegen das vorgeschlagene Verhalten entscheiden. Charakteristisch für Nudging ist somit die Wahrung der Autonomie, während gleichzeitig eine positive Verhaltensänderung angestrebt wird. Typische Beispiele sind Erinnerungen, Hinweise oder Vorschläge, die zu einem bestimmten Zeitpunkt abhängig vom Kontext gegeben werden [13].

Das im Rahmen dieser Arbeit entwickelte Assistenzsystem nutzt Nudging als zentrales Interaktionsprinzip. Anstatt verbindliche Handlungsanweisungen vorzugeben, werden Push-Benachrichtigungen als dezente Impulse eingesetzt, um den Nutzer zu einer regelmäßigeren Flüssigkeitsaufnahme oder zu mehr Bewegung zu motivieren. Die Auslösung dieser Benachrichtigungen erfolgt kontextabhängig auf Basis automatisch erfasster Daten.

2.7 Verwandte Gesten

Im Alltag kommen verschiedene Bewegungen vor, die ein ähnliches Bewegungsmuster wie Trinkgesten aufweisen. Meist handelt es sich dabei um Bewegungen, bei denen die Hand oder der Arm in Richtung Gesicht geführt wird. Zwar unterscheiden sich diese Bewegungen meist in mindestens einem Aspekt von echten Trinkgesten, wie z.B. der Abwesenheit

einer Haltephase vor dem Mund oder der fehlenden Rotationsbewegung des Handgelenks. Trotzdem können diese Bewegungen zu Fehlklassifikationen führen und oft nur durch gute, vielfältige Trainingsdaten von echten Trinkgesten unterschieden werden.

Beispiele für solche verwandten Gesten sind:

- Essen: sehr ähnlicher Ablauf, aber ohne die Rotationsbewegung des Handgelenks
- Telefonieren: ähnliche Armbewegung Richtung Kopf, aber dann keine Haltephase vor dem Mund, sondern am Ohr
- Gesicht berühren/kratzen: schnelle, kurze Bewegungen ohne Haltephase
- Rauchen/Dampfen: sehr ähnlicher Ablauf, aber ohne Rotationsbewegung
- Gegenstände mit einer Hand anheben: gleiche Armbewegung zum Greifen des Objekts, keine Bewegung zum Mund und keine Haltephase, gleiche Bewegung beim Abstellen

2.8 Verwandte Arbeiten

2.8.1 Trinkgestenerkennung mit am Handgelenk getragener IMU

In dem Paper „Drinking Gesture Detection Using Wrist-Worn IMU Sensors with Multi-Stage Temporal Convolutional Network in Free-Living Environments“ untersuchen Wang et al. die automatische Erkennung von Trinkgesten anhand von Daten einer am Handgelenk getragenen Inertialen Messeinheit (IMU) [21]. Ziel dieser Arbeit ist es zu evaluieren, inwieweit Trinkgesten in realistischen Alltagsszenarien zuverlässig erkannt werden können, ohne auf zusätzliche Sensorik oder stark kontrollierte Umgebungen angewiesen zu sein.

Für die Datenerhebung wurde der Shimmer3 verwendet, eine spezialisierte Forschungs-IMU, die Beschleunigungs- und Gyroskopdaten liefert. Für die Klassifikation der Sensordaten setzen die Autoren ein Multi-Stage Temporal Convolutional Network (MS-TCN) ein [3]. Ziel ist es, nicht nur einzelne Zeitpunkte, sondern vollständige Trinkgesten über deren Dauer hinweg robust zu erkennen. Die Autoren verfolgen dabei bewusst den Ansatz eines personenunabhängigen Modells, das auf neue Nutzer generalisieren kann.

Die Evaluation erfolgte mithilfe des Leave-One-Subject-Out-Verfahrens (LOSO), um die Generalisierungsfähigkeit des Modells zu bewerten. Auf dem Datensatz aus den „teilweise kontrollierten“ Umgebungen erreichte das MS-TCN einen F1-Score von 0.943. Im „freien Lebensumfeld“ konnte ein F1-Score von 0.90 erzielt werden, was die Eignung des Ansatzes auch unter realistischen Alltagsbedingungen belegt.

Zum Vergleich wurde ein Convolutional Neural Network mit Long Short-Term Memory (CNN-LSTM) eingesetzt, ein häufig verwendeter Ansatz zur Verarbeitung zeitlicher Sensordaten. Das MS-TCN zeigte in beiden Szenarien eine deutliche Leistungssteigerung. Auf dem „teilweise kontrollierten“ Datensatz erreichte das CNN-LSTM nur einen F1-Score von 0.88 und auf dem „freien Lebensumfeld“ Datensatz nur einen F1-Score von 0.799. Diese Ergebnisse unterstreichen die besondere Eignung von MS-TCN-Modellen für die robuste Erkennung komplexer Gesten in Alltagsdaten.

Die Arbeit von Wang et al. zeigt, dass Trinkgesten zuverlässig allein anhand von IMU-Daten am Handgelenk erkannt werden können und dass Multi-Stage-Ansätze insbesondere bei variablen Bewegungsabläufen deutliche Vorteile bieten. Die Ergebnisse bilden damit eine zentrale Grundlage für die vorliegende Arbeit, in der ein vergleichbarer Ansatz auf Smartwatch-Sensorik übertragen und zur Ableitung des Hydrationsstatus im Alltag älterer Menschen eingesetzt wird.

2.8.2 Becherbasierte Systeme mit Lastsensor oder integriertem IMU

Mehrere Arbeiten verfolgen einen hardwaregestützten Ansatz zur automatisierten Erfassung der Flüssigkeitsaufnahme, indem sie Trinkereignisse detektieren und die getrunkene Menge direkt messen.

Soubam et al. kombinieren eine Arduino-basierte Sockel-Einheit mit Lastsensor (zur Messung der Volumenänderung), einem Beschleunigungsmesser und Bluetooth mit einer Smartwatch, die über ihren Beschleunigungssensor Trinkereignisse als „Trinken ja/nein“ detektiert und nur bei erkannten Trinkgesten die Volumenänderung des Sockels auswertet, wodurch Verschütten oder das Verfälschen der Messung durch Trinken anderer Personen weitgehend ausgeschlossen werden kann. Für ihre Evaluation ließen sie 11 Probanden kontinuierlich Schlücke trinken und testeten unterschiedliche Klassifikatoren (Decision Tree, Naive Bayes, Random Forest, SVM). Der Random-Forest-Klassifikator erreichte dabei einen F1-Score von 95,97% [15].

Liu et al. verfolgen einen ähnlichen Zweck mit einem 3D-gedruckten „Smart Cup“, in dessen Boden eine OPAL-IMU integriert ist, wovon allerdings nur der Beschleunigungsmesser verwendet wird. Auf ihrem Datensatz mit 11 Probanden (472,8 Minuten, 132 Trinkepisoden) erzielte ein kNN-Modell mit der Fenstergröße von 4 Sekunden die beste Leistung mit einem segmentalen F1-Score von 89,92% bzw. frame-wise 85,88% [9].

Beide Ansätze liefern gute Erkennungs- und Messgenauigkeiten, teilen aber den praktischen Nachteil, dass Nutzer stets spezielle, kalibrierte Gefäße oder Sockel verwenden müssen. Das in dieser Arbeit entwickelte System grenzt sich von diesem Ansatz ab, indem es ausschließlich auf die Sensorik einer handelsüblichen Smartwatch setzt, ohne dass zusätzliche Hardware erforderlich ist.

3 Design und Implementierung

Das in dieser Arbeit entwickelte Assistenzsystem besteht aus mehreren Komponenten, die zusammenarbeiten, um Rückschlüsse auf den Hydrations- und Bewegungsstatus des Nutzers zu geben. Dazu werden Tagesdaten über eine Smartwatch erfasst, an einen Server übertragen und dort analysiert. Tagesdaten sind Daten, die im Alltag eines Nutzers erfasst werden. Bei Unterschreitung von definierten Mindestwerten werden kontextabhängige Push-Benachrichtigungen an die Smartwatch gesendet, um den Nutzer zu mehr Trinken oder Bewegung zu motivieren.

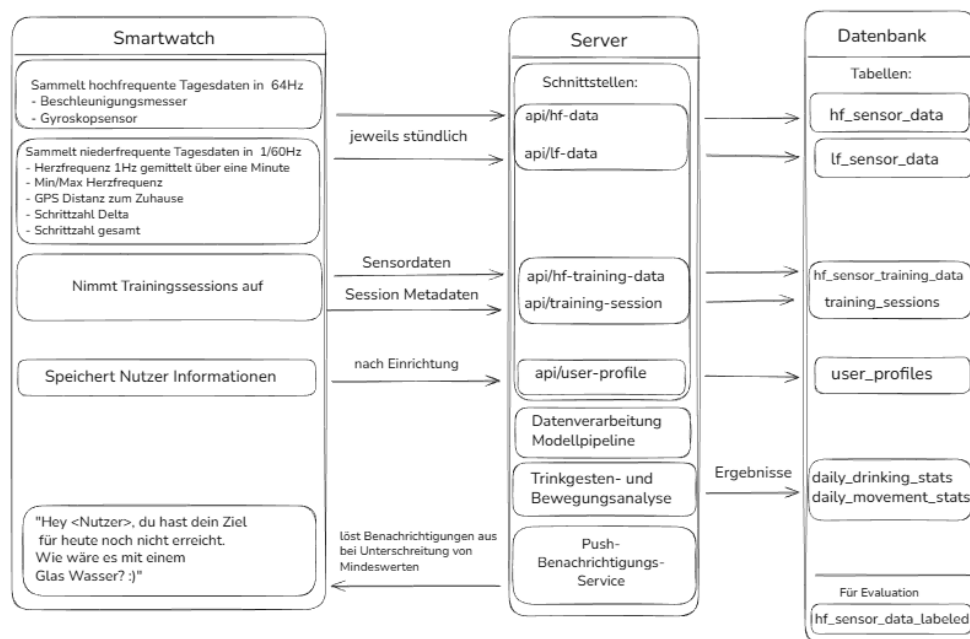


Abbildung 3.1: Übersicht der Systemkomponenten

3.1 PC-Spezifikationen, Tools und Projektumgebungen

Das in dieser Arbeit eingesetzte System (PC) hat folgende Spezifikationen:

- Prozessor: AMD Ryzen 7 5700X3D @ 3 GHz
- Arbeitsspeicher: 16GB DDR4
- Grafikkarte: NVIDIA GeForce RTX 3060 TI 8 GB
- Betriebssystem: Windows 11

Details zu Bibliotheken, Tools und Umgebungen:

- Tools: Android Studio, PyCharm, Copilot
- Programmiersprachen: Kotlin (2.0.21 - JVM 17), Python (3.11.9)
- Pixelwatch 1 Betriebssystem: Wear OS 5.1
- Wichtigste Kotlin-Bibliotheken: Android SDK 35, Wear Compose (1.2.1), Retrofit (2.9.0), OkHttp3 (4.11.0), Room (2.6.1)
- Wichtigste Python-Bibliotheken: Flask (3.0.0), NumPy (2.3.4), PyTorch (2.9.0), pytz (2024.1), Matplotlib (3.5.0), psycopg2 (2.9.9)
- Datenbank: TimescaleDB (Erweiterung für PostgreSQL 18)

Copilot wurde während der Entwicklung des Systems für partielle Codegenerierungen und -vervollständigungen und während des Schreibens der Arbeit für Grammatikprüfungen eingesetzt. Sämtliche Inhalte, Argumentationen, Ergebnisse und Schlussfolgerungen stammen vom Autor.

3.2 Anforderungen

In den Tabellen 3.1 und 3.2 werden die Anforderungen an das Assistenzsystem in funktionale und nicht-funktionale Anforderungen unterteilt.

Tabelle 3.1: Funktionale Anforderungen

ID	Anforderung
F1	Kontinuierliche Erfassung von Sensordaten (Beschleunigung, Gyroskop, Herzfrequenz, Schrittzähler, GPS) auf einer Smartwatch mit dem Betriebssystem Wear OS 5
F2	Der Nutzer muss Trainingssessions durchführen können, um Trinkgesten für das spätere Modelltraining aufzuzeichnen
F3	Das System soll auf Basis der aufgenommenen Trainingssessions ein individuelles Modell zur Trinkgestenerkennung trainieren können
F4	Es dürfen keine Daten verloren gehen, wenn die Smartwatch zum Zeitpunkt des Uploads keine Internetverbindung hat
F5	Das System soll den Hydrations- und Bewegungsstatus des Nutzers automatisch zu festen Zeitpunkten analysieren können
F6	Der Server soll kontextabhängige Push-Benachrichtigungen senden können, wenn Mindestwerte für Trinken oder Bewegung unterschritten werden.

Tabelle 3.2: Nicht-funktionale Anforderungen

ID	Anforderung
NF1	Die Smartwatch sollte möglichst energieeffizient arbeiten, um eine lange Akkulaufzeit zu gewährleisten
NF2	Es sollen keine personenbezogenen Daten gespeichert oder übertragen werden, die zur Identifizierung der Nutzer führen könnten
NF3	Die Erkennung soll robust gegenüber verschiedenen Behältern und Umgebungen sein
NF4	Das System soll minimalinvasiv im Alltag der Nutzer funktionieren und nach Einrichtung keine aktive Interaktion erfordern
NF5	Die Smartwatch sollte möglichst lange getragen werden können, ohne ständig aufgeladen werden zu müssen
NF6	Es sollen nur Sensordaten gesammelt werden, wenn die Smartwatch getragen wird, um den Akku nicht unnötig zu belasten

3.3 Trainingsmodus

Die Daten für das Modelltraining werden über den Trainingsmodus der entwickelten Smartwatch-Applikation von jedem Nutzer selbst aufgenommen. In diesem Modus werden die Daten des Beschleunigungsmessers und Gyroskopsensors in vier Trainingssessions à 30 Minuten mit einer Trinkgeste alle 2-3 Minuten aufgenommen. Während der Trainingssessions werden die Trinkgesten manuell durch den Nutzer auf der Smartwatch gekennzeichnet. Er muss dabei vor jeder Trinkgeste die Aufnahme starten und nach der Trinkgeste wieder stoppen.

Die Trinkgesten sollen möglichst natürlich ausgeführt werden und können aus beliebigen Behältern (Tasse, Flasche, Glas etc.) erfolgen. Am besten werden alle Behälter verwendet, aus denen der Nutzer im Alltag trinkt. Die Trinkgesten sollen mal einen oder mehrere Schlücke umfassen und in verschiedenen Umgebungen (z.B. sitzend am Tisch, stehend in der Küche) erfolgen. Je größer die Varianz der aufgenommenen Trinkgesten ist, desto robuster kann das spätere Modell die Trinkgesten im Alltag erkennen. Am Ende jeder Trainingssession gibt der Nutzer die getrunkene Menge in Milliliter ein. Im Anschluss werden die Daten über den Server in die Datenbank geschrieben.

Alle in dieser Arbeit verwendeten Daten wurden mit einer Google Pixelwatch 1 Smartwatch mittels der Smartwatch-Anwendung vom Autor selbst aufgenommen. Die nachfolgende Beschreibung enthält die Details zu den aufgenommenen Trainings- und Evaluationsdaten.

3.3.1 In dieser Arbeit verwendete Trainingsdaten

Es wurden vier Trainingssessions aufgenommen, mit einer Gesamtdauer von 121 Minuten und 30 Sekunden mit insgesamt 50 gekennzeichneten Trinkgesten. Die Gesamttrinkdauer über alle Sessions beträgt 5 Minuten und 9 Sekunden und das Verhältnis von Trinkgesten zu Nicht-Trinkgesten beträgt ca. 1:24. Die Hälfte der Trinkgesten erfolgte aus einer Tasse und die andere Hälfte aus einer Glasflasche. Manche Trinkgesten bestanden aus einem Schluck, andere aus mehreren Schlücken. Außerdem wurden die Trinkgesten in verschiedenen Positionen und Umgebungen ausgeführt: sitzend und stehend am Schreibtisch, in der Küche und im Wohnzimmer.

In drei der vier Sessions wurde 750 ml, in der vierten Session 1000 ml getrunken. Die durchschnittliche Trinkdauer einer Session beträgt 77 Sekunden. Die durchschnittliche Trinkmenge pro Session beträgt 812,5 ml.

Um die Mindesttrinkdauer zu berechnen, um die empfohlene tägliche Flüssigkeitszufuhr von 2000 ml zu erreichen, wird folgende Formel verwendet:

$$\text{Mindesttrinkdauer} = \frac{\text{durchschnittliche Trinkdauer pro Session}}{\text{durchschnittliche Trinkmenge pro Session}} \times 2000 \quad (3.1)$$

Für Frauen wird mit 1600 multipliziert.

Die Mindesttrinkdauer für den Autor, um 2000 ml zu erreichen, beträgt somit ca. 190 Sekunden bzw. 3 Minuten und 10 Sekunden pro Tag und gilt nur für den Autor und kann nicht auf andere Personen übertragen werden.

3.3.2 Evaluationsdaten

Die Evaluationsdaten enthalten gekennzeichnete Tagesdaten über 5 Tage mit einer Gesamtdauer von 23 Stunden und 20 Minuten. Die Gesamtdauer der gekennzeichneten Trinkgesten beträgt 673,4 Sekunden mit einem Verhältnis von Trinkgesten zu Nicht-Trinkgesten von 1:127. Die Daten wurden mit der entwickelten Smartwatch-Applikation erfasst und gekennzeichnet und enthalten Sensordaten des Beschleunigungsmessers und Gyroskopsensors. Die Trinkereignisse wurden manuell durch den Autor vor jeder Trinkaktion auf der Smartwatch gekennzeichnet. Diese Daten wurden ausschließlich zur Evaluation der Trinkgestenerkennung dieser Arbeit aufgenommen und sind nicht Teil der verwendeten Trainingsdaten.

3.4 Systemarchitektur

Das entwickelte Assistenzsystem besteht aus mehreren miteinander verknüpften Komponenten, die gemeinsam die Erfassung, Analyse und Rückmeldung des Gesundheitsstatus ermöglichen. Die Smartwatch-Anwendung wurde für eine Google Pixelwatch 1 mit dem Wear OS 5 Betriebssystem entwickelt und nutzt die integrierten Sensoren zur kontinuierlichen Erfassung von Bewegungsdaten, welche über einen Server an eine Datenbank übertragen werden. Der Server kümmert sich um die Datenverarbeitung, enthält die Pipeline zur Erstellung des Modells, stellt die Schnittstellen bereit, integriert externe Wetterdaten

und löst Push-Benachrichtigungen, bei Unterschreitung von festgelegten Mindestwerten, aus.

3.4.1 Datenbank

Das System verwendet eine TimescaleDB-Datenbank zur Speicherung aller aufkommen- den Daten. TimescaleDB ist eine Open-Source Erweiterung für PostgreSQL, die für die effiziente Speicherung und Abfrage von Zeitreihen optimiert ist [19].

Auf der Smartwatch wird eine lokale Room-Datenbank verwendet, um die erfassten Sensordaten zwischenspeichern, bevor sie stündlich an den Server übertragen werden. Dies ist wichtig, da innerhalb einer Stunde über 460 000 Sensordaten erfasst und gespeichert werden und diese sonst verloren gehen würden, wenn die Anwendung geschlossen oder die Smartwatch neu gestartet wird. Außerdem kann es passieren, dass der RAM voll läuft, wenn die Daten nicht zwischengespeichert werden.

3.4.2 Smartwatch-Applikation

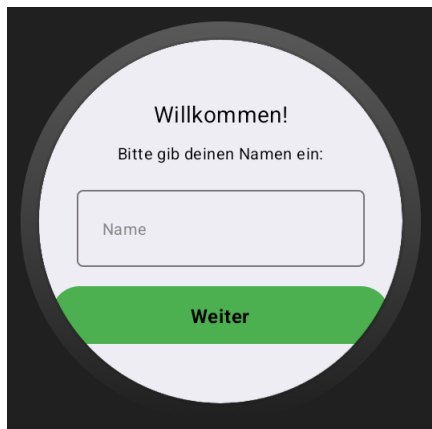
Die Smartwatch-Applikation wurde in Kotlin geschrieben und speziell für das Wear OS 5 Betriebssystem entwickelt. Sie ist verantwortlich für die kontinuierliche Erfassung der Sensordaten über die integrierten Sensoren sowie für die stündliche Übertragung dieser Daten an den Server. Wear OS 5 bietet Entwicklern vollen Zugriff auf die Sensor-APIs und das Einrichten von Hintergrunddiensten, wodurch eine kontinuierliche Datenerfassung auch bei inaktiver Nutzung der Smartwatch möglich ist. Die Applikation besteht aus zwei Modulen, die im Folgenden näher beschrieben werden.

Einrichtung

Wenn der Nutzer die Smartwatch-Anwendung zum ersten Mal startet, muss er bestimmte Berechtigungen erteilen, damit die Applikation korrekt funktioniert. Hierzu gehört der Zugriff auf die Sensoren (Beschleunigungsmesser, Gyroskop, Herzfrequenzsensor, Schrittzähler, GPS) und die Erlaubnis, im Hintergrund zu laufen. Außerdem muss der Nutzer der App erlauben, Push-Benachrichtigungen zu empfangen, da diese für die Nudging-Funktionalität benötigt werden.

Danach wird er darauf hingewiesen, die Einrichtung durchzuführen. Zuerst gibt der Nutzer einen Namen an. Danach setzt er sein Zuhause als Standort per Knopfdruck in der Applikation. Aus den Standortdaten wird ausschließlich die Stadt des Nutzers in seinem Nutzerprofil gespeichert. Die genauen Standortdaten werden nur lokal auf der Smartwatch gespeichert. Dadurch kann die Smartwatch bei der Sammlung der Tagesdaten die Distanz zum Zuhause berechnen und an den Server übertragen, ohne dass genaue Standortdaten gespeichert werden müssen. Die Stadt wird für die wetterabhängige Bewegungserinnerung verwendet.

Im nächsten Schritt gibt der Nutzer sein Geschlecht an, das zur Bestimmung des Hydrationsstatus dient, sowie sein Alter, das für die Berechnung der Aktivitätsminuten benötigt wird.



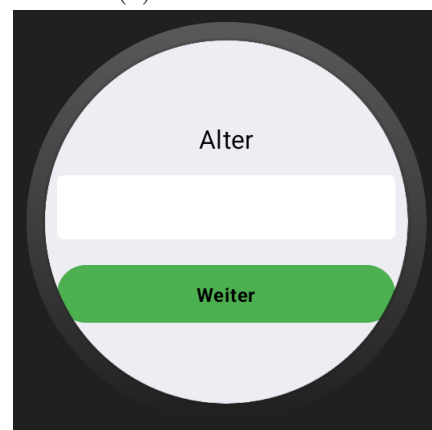
(a) Namen eingeben



(b) Zuhause setzen



(c) Geschlecht eingeben



(d) Alter angeben

Abbildung 3.2: Einrichtungsschritte der Smartwatch-Applikation

Aufnahme der Trainingsdaten

Der letzte und wichtigste Schritt der Einrichtung ist das Durchführen von Trainingssessions zur Aufnahme von Trinkgesten. Jeder Nutzer muss mindestens 4 Trainingssessions durchführen, bevor die Analyse der Tagesdaten auf Trinkgesten aktiviert wird. Eine möglichst große Vielfalt an Trinkgesten während der Trainingssessions erhöht die Erkennungsrate im Alltag. Stehend, sitzend, verschiedene Behälter (Tasse, Glas, Flasche) und verschiedene Trinkgeschwindigkeiten sollten dabei abgedeckt werden. Außerdem soll der Nutzer messen, wie viel er während der Trainingssessions getrunken hat.

Eine Trainingssession dauert 30 Minuten und der Nutzer wird dazu aufgefordert, alle 2-3 Minuten einen oder mehrere Schlucke aus einem Behälter zu trinken. Je größer die Vielfalt der Trinkgesten während der Trainingssession ist, desto erfolgreicher kann das Modell später Trinkgesten im Alltag erkennen. Während der Trainingssession werden die Sensordaten des Beschleunigungsmessers und des Gyroskops zusammen mit Zeitstempeln und Kennzeichnung der Trinkereignisse gespeichert.

Der genaue Ablauf einer Trainingssession und das Kennzeichnen einer Trinkgeste laufen folgendermaßen ab:

- Der Nutzer startet die Trainingssession auf der Smartwatch
- Wenn er trinken möchte, drückt er auf den „Trinkgeste aufnehmen“-Button auf der Smartwatch
- Alle folgenden erfassten Daten werden als Trinkgeste markiert und der Text des „Trinkgeste aufnehmen“-Buttons ändert sich zu „Trinkgeste abschließen“
- Nachdem der Nutzer fertig getrunken hat, drückt er auf den „Trinkgeste abschließen“-Button
- Alle folgenden erfassten Daten werden wieder als Nicht-Trinken markiert und der Text des „Trinkgeste abschließen“-Buttons ändert sich zurück zu „Trinkgeste aufnehmen“
- Nach 30 Minuten endet die Trainingssession und der „Speichern“-Button ersetzt den „Trinkgeste aufnehmen“/„Trinkgeste abschließen“-Button

- Der Nutzer drückt auf den „Speichern“-Button, gibt die getrunkene Menge in Milliliter ein und dann werden die Trainingsdaten über den Server an die Datenbank übertragen



Abbildung 3.3: Kennzeichnen einer Trinkgeste während einer Trainingssession

Sind diese Schritte abgeschlossen, wird ein Modell für die Trinkgestenerkennung auf den aufgenommenen Trainingsdaten trainiert. Sobald ein Modell existiert, werden die Tagesdaten automatisch im Hintergrund erfasst und stündlich an den Server übertragen.

Sollte die Performance der Trinkgestenerkennung im Alltag nicht zufriedenstellend sein, kann der Nutzer jederzeit weitere Trainingssessions durchführen, um das Modell mit zusätzlichen Daten zu verbessern.

Sammlung der Tagesdaten

Dieses Modul ist für die kontinuierliche Erfassung der Sensordaten im Alltag zuständig. Es nutzt die Sensor-APIs von Wear OS 5, um Daten vom Beschleunigungsmesser, Gyroskop, Herzfrequenzsensor, Schrittzähler und GPS-Sensor zu sammeln.

Die Daten des Beschleunigungsmessers und Gyroskopsensors bilden die Grundlage für die Trinkgestenerkennung und werden mit einer Abtastrate von 64 Hz erfasst und als

Rohdaten übertragen. Durch die Wahl einer Abtastrate von 64 Hz wird ein guter Kompromiss zwischen Datenqualität, Datengröße und Energieverbrauch erzielt. Im Folgenden werden diese Daten als hochfrequente Sensordaten bezeichnet.

Die Daten des Herzfrequenzsensors, Schrittzählers und GPS-Sensors werden für die Bewegungsanalyse verwendet. Die Herzfrequenz wird mit 1 Hz erfasst und über eine Minute gemittelt, zusätzlich werden die minimale und maximale Herzfrequenz pro Minute gesammelt. Die Schrittzahl pro Minute zusammen mit der Gesamtanzahl der Schritte des Tages wird ebenfalls gespeichert. Die GPS-Daten werden als Distanz zum festgelegten Zuhause in Metern gespeichert. Dies schützt die Privatsphäre der Nutzer, da keine genauen Standortdaten übertragen werden. Außerdem wird die Bewegungsgeschwindigkeit in km/h gespeichert, um Rückschlüsse auf die Intensität der Bewegung ziehen zu können. Im Folgenden werden diese Daten als niederfrequente Sensordaten bezeichnet.

Die erfassten Sensordaten werden lokal in einer Room-Datenbank zwischengespeichert und stündlich an die TimescaleDB übertragen.

3.4.3 Server

Der Server wurde in Python implementiert und übernimmt mehrere zentrale Aufgaben des Assistenzsystems. Er enthält die Pipeline zur Erstellung des Modells für die Trinkgestenerkennung, stellt die APIs für die Smartwatch-Applikation bereit, integriert externe Wetterdaten und löst Push-Benachrichtigungen aus, wenn Mindestwerte für Trinken oder Bewegung unterschritten werden. Er führt außerdem die Analyse des Hydrations- und Bewegungsstatus durch und speichert alle relevanten Daten in der TimescaleDB.

ModellPipeline

Die Modell-Pipeline trainiert ein MS-TCN Modell mit den aufgenommenen Trainingssessions des jeweiligen Nutzers. Jeder Nutzer hat ein eigenes Modell, das auf seine individuellen Trinkgewohnheiten trainiert wird. Im Folgenden werden die einzelnen Schritte der Pipeline beschrieben.

1. Datenvorbereitung

Um die aufgezeichneten Trainingssessions für das Modelltraining zu verwenden, werden sie zuerst aus der Datenbank geladen und in das binäre Serialisierungsformat Pickle

(PKL) konvertiert. Vor der Konvertierung werden die Sensordaten zuerst von 64 Hz, via linearer Interpolation, auf 16 Hz herunter skaliert. Dies reduziert die Datenmenge, verkürzt die Rechenzeit, glättet Rauschen und ist trotzdem ausreichend, um Bewegungsmuster zu erkennen.

Dann wird für jede Trainingssession jeder der sechs Sensordatenkanäle separat mithilfe des Z-Scores normalisiert, um verschiedene Einheiten und Größenordnungen zu vereinheitlichen. Für jeden der sechs Sensordatenkanäle (3 Achsen Beschleunigung, 3 Achsen Gyroskop) werden Mittelwert (μ) und Standardabweichung (σ) über alle Frames der Session berechnet. Die Kanäle werden anschließend mit $x' = (x - \mu)/\sigma$ skaliert, sodass jeder Kanal einen Mittelwert von ungefähr 0 und eine Standardabweichung von ungefähr 1 aufweist.

Jede PKL enthält die normalisierten Sensordaten als Features, sowie die zugehörigen Labels, die kennzeichnen, ob zu einem bestimmten Zeitpunkt eine Trinkgeste (Label 1) oder keine Trinkgeste (Label 0) stattgefunden hat.

2. Modellarchitektur

Das verwendete Multi-Stage Temporal Convolutional Network (MS-TCN) besteht aus mehreren aufeinanderfolgenden Stages, bestehend aus „dilated temporal convolutions“, die erste Vorhersagen generieren und diese in den nachfolgenden Stages verfeinern. Es ist dadurch besonders gut geeignet, um Bewegungsmuster in Zeitreihendaten zu erkennen [3].

Die konkrete Konfiguration der Hyperparameter des Modells ist wie folgt:

- **Features:** 6 (3 Achsen Beschleunigung + 3 Achsen Gyroskop)
- **Anzahl der Ausgabeklassen:** 2 (Trinken/Nicht-Trinken)
- **Anzahl der Stufen:** 4
- **Anzahl der Layer pro Stufe:** 10
- **Anzahl der Filter pro Layer:** 64
- **Anzahl der Epochen:** 100
- **Kernel-Größe:** 3
- **Dropout-Rate:** 0.3

- **Batch size:** 16
- **Lernrate:** 0.0005
- **Early-stopping:** 20

3. Modelltraining Es wird ein MS-TCN Modell auf dem gesamten Datensatz über 100 Epochen trainiert. Eine Epoche bezeichnet dabei einen vollständigen Durchlauf durch den gesamten Trainingsdatensatz. Wie im Paper von Wang et al. wird die Loss-Funktion (Cross-Entropy + truncated MSE smoothing) zusammen mit dem Adam-Optimizer eingesetzt, um die Modellperformance während des Trainings zu messen [8, 21]. Anhand dessen wird entschieden, welches Modell nach jeder Epoche ausgewählt wird.

Um die Menge an Trainingsbeispielen zu erhöhen, werden die Trainingsdaten in überlappende Sequenzen der Länge 1024 aufgeteilt. Die Überlappungsrate der Sequenzen beträgt 93.5%.

Das Modelltraining dauert auf dem angegebenen System etwa 6 Minuten, und das trainierte Modell ist etwa 5 MB groß.

Die Diskussion zur Wahl der Konfiguration der Sequenzlänge und Überlappungsrate erfolgt im Kapitel 4.1.

APIs

Der Server stellt eine Vielzahl an APIs bereit, die von der Smartwatch-Applikation genutzt werden. Dazu gehören folgende Endpunkte:

- **zum Hochladen der hochfrequenten Tagesdaten, der niederfrequenten Tagesdaten, der gekennzeichneten Trainingssessions und der gekennzeichneten Tagesdaten für die Evaluation**
- **zum Abrufen und Setzen der Nutzerprofile**
- **zum Abrufen der Anzahl der durchgeführten Trainingssessions**
- **für Statistiken und Auswertungen der Tagesdaten**
- **zum Bestätigen des Empfangs von Bewegungs- oder Trinkerinnerungen**
- **zum Starten der Analyse der Tagesdaten auf Trinkgesten**

- zum manuellen Abfragen des Hydrations- und Bewegungsstatus durch den Nutzer
- zum Überprüfen, ob eine Verbindung zum Server besteht

Push-Benachrichtigungs-Service

Sollten nach der Analyse der Tagesdaten die Mindestwerte für Trinken oder Bewegung nicht erreicht worden sein, sendet der Server eine kontextabhängige Push-Benachrichtigung an die Smartwatch.

Wetterdienst Integration

Der API Server integriert externe Wetterdaten, um die Bewegungserinnerungen an das Wetter anzupassen. So kann der Server bei der Analyse der Tagesdaten berücksichtigen, ob das Wetter aktuell oder in naher Zukunft für Outdoor-Aktivitäten geeignet ist.

3.5 Implementierungsdetails

3.5.1 Ablauf der Analyse der Tagesdaten auf Trinkgesten

Die Analyse der Tagesdaten auf Trinkgesten erfolgt viermal täglich zu festen Zeitpunkten: 12 Uhr, 15 Uhr, 18 Uhr und 21 Uhr. Dies ermöglicht es, den Nutzer regelmäßig an das Trinken zu erinnern und stellt so eine kontinuierliche Hydration über den Tag sicher. Zu diesen Zeitpunkten werden alle bisher erfassten Sensordaten des Beschleunigungsmessers und Gyroskops mithilfe des trainierten MS-TCN Modells ausgewertet.

Das Ergebnis der Auswertung wird in die Datenbank geschrieben und enthält die Anzahl der erkannten Trinkgesten und deren Gesamtdauer.

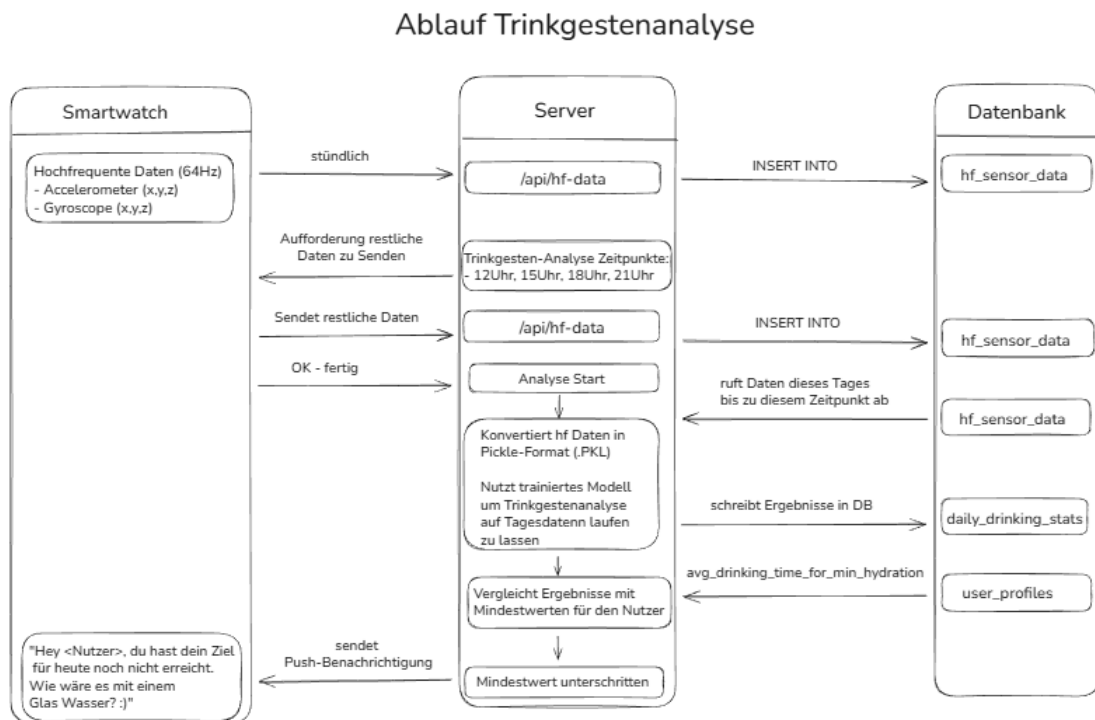


Abbildung 3.4: Ablaufdiagramm zur Analyse der Tagesdaten auf Trinkgesten

Auslösen der Trinkerinnerung

Die empfohlenen Mindestwerte basieren auf den ESPEN-Richtlinien für die Flüssigkeitsaufnahme im Alter. Für Männer wird ein Mindestwert für die Flüssigkeitszufuhr von 2000 ml und für Frauen ein Mindestwert von 1600 ml pro Tag angesetzt [20]. Über die Trainingssessions des jeweiligen Nutzers wird die individuelle Mindesttrinkdauer berechnet, die benötigt wird, um die empfohlene Tagesmenge zu erreichen. Die Trinkdauer wird dabei gleichmäßig auf die vier Zeitpunkte verteilt, sodass zu jedem Zeitpunkt das jeweilige Viertel der Tagesmenge erreicht sein sollte. Ist die Mindesttrinkdauer eines Nutzers beispielsweise 200 Sekunden, dann sollte er bis 12 Uhr 50 Sekunden, bis 15 Uhr 100 Sekunden, bis 18 Uhr 150 Sekunden und bis 21 Uhr 200 Sekunden getrunken haben. Wenn die bisherige Trinkdauer unter dem jeweiligen Mindestwert liegt, wird eine Push-Benachrichtigung ausgelöst, die den Nutzer daran erinnert, etwas zu trinken.



Abbildung 3.5: Beispiel Push-Benachrichtigung für eine Trinkerinnerung

3.5.2 Ablauf der Analyse der Tagesdaten auf Bewegung

Die Analyse der Tagesdaten auf Bewegung erfolgt zweimal täglich, um 13 Uhr und um 18 Uhr. Zu diesen Zeitpunkten werden die bisher erfassten Sensordaten des Herzfrequenzsensors, Schrittzählers und GPS-Sensors ausgewertet und die körperliche Aktivität des Tages analysiert. Das Ergebnis der Auswertung wird in die Datenbank geschrieben und enthält die Anzahl an Aktivitätsminuten.

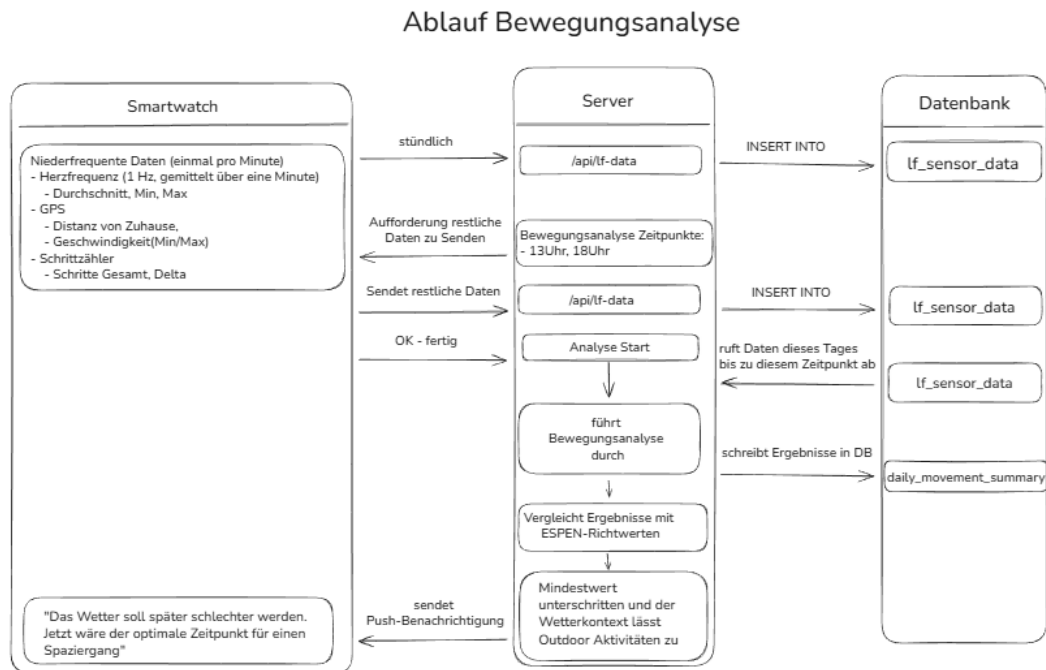


Abbildung 3.6: Ablaufdiagramm zur Analyse der Tagesdaten auf Bewegung

Auslösen der Bewegungserinnerung

Für Personen über 65 Jahre empfiehlt die WHO zwischen 150 und 300 Minuten moderate körperliche Aktivität pro Woche. Für diese Arbeit wird ein Mindestwert von 30 Minuten pro Tag angesetzt.

Um die Aktivitätsminuten zu berechnen, wird in dieser Arbeit für eine moderate körperliche Aktivität eine minimale Herzfrequenz von 50% der maximalen Herzfrequenz angenommen. Die maximale Herzfrequenz wird dabei mit der Formel

$$\text{Herzfrequenz}_{\max} = 208 - 0.7 \cdot \text{Alter} \quad (3.2)$$

berechnet [18].

Die bisherigen Aktivitätsminuten des Tages werden mit den empfohlenen Mindestwerten der WHO verglichen, um zu entscheiden, ob eine Push-Benachrichtigung ausgelöst werden soll. Dabei wird auch das aktuelle und vorhergesagte Wetter berücksichtigt. Wenn die bisherigen Aktivitätsminuten unter dem jeweiligen Mindestwert liegen und das Wetter

aktuell oder in naher Zukunft für Outdoor-Aktivitäten geeignet ist, wird eine Bewegungserinnerung ausgelöst. Ist das Wetter aktuell nicht geeignet, wird keine Erinnerung ausgelöst.

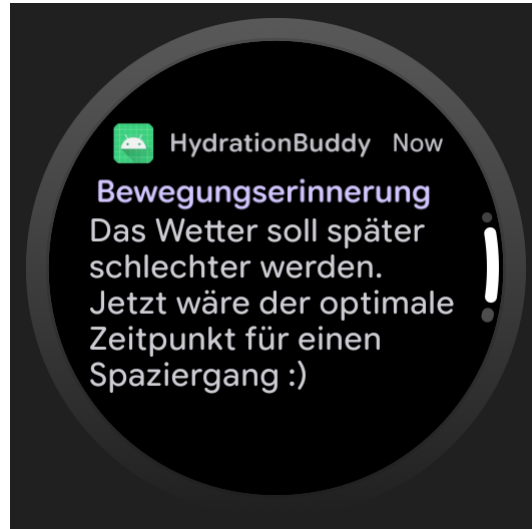


Abbildung 3.7: Beispiel Push-Benachrichtigung für eine Bewegungserinnerung

3.6 Datenübertragung

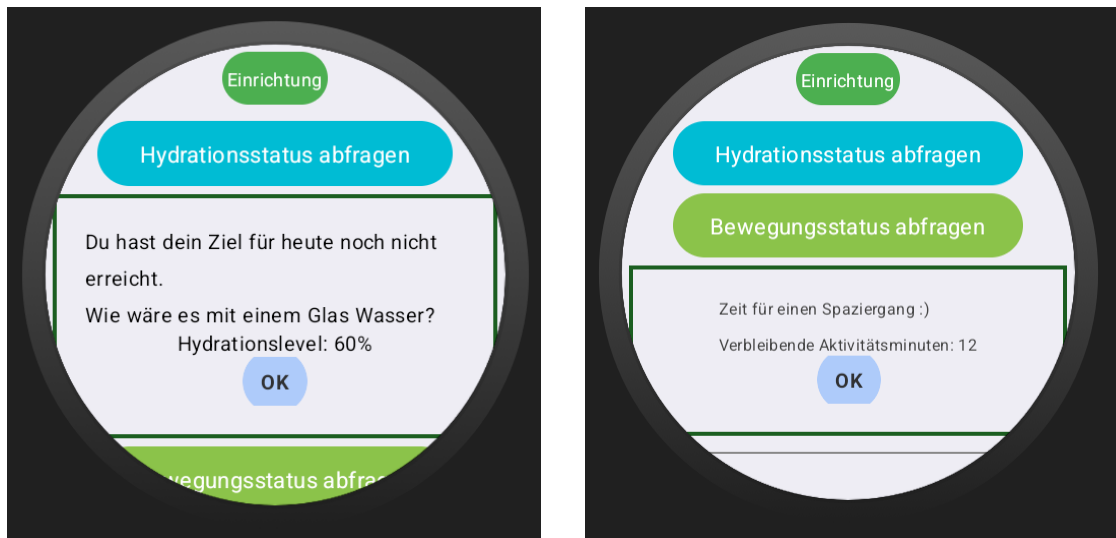
Die Datenübertragung der Smartwatch an den Server erfolgt stündlich oder wenn der Nutzer seinen Hydrations- oder Bewegungsstatus manuell anfordert. Sollte die Smartwatch mit einem Smartphone verbunden sein, erfolgt die Datenübertragung über das Bluetooth Low Energy (BLE) Protokoll an das Smartphone, welches die Daten dann über das Mobilfunknetz oder eine bestehende WLAN-Verbindung an den Server sendet. Ist keine Verbindung zu einem Smartphone vorhanden, sendet die Smartwatch die Daten direkt über eine Netzwerkverbindung an den Server, insofern eine besteht. Android entscheidet dabei automatisch, welche der beiden Methoden verwendet wird, priorisiert aber immer die Bluetooth Low Energy Verbindung.

Sollte zum Zeitpunkt der Datenübertragung keine der Methoden verfügbar sein, werden die Daten übertragen, sobald eine Verbindung verfügbar ist.

Vor jedem Upload werden die Daten mit gzip komprimiert. Hiermit wird die Datenmenge im Schnitt um 70% reduziert und beschleunigt damit die Datenübertragung, besonders bei langsamen Netzwerkverbindungen.

3.7 Zusatzfunktionen

Nutzer können jederzeit ihren aktuellen Hydrations- und Bewegungsstatus manuell über die Smartwatch anfordern. Hierzu sendet die Smartwatch eine Anfrage an den Server, der dann die bisher erfassten Tagesdaten analysiert und den aktuellen Status zurücksendet.



(a) Manuelle Abfrage des Hydrationsstatus

(b) Manuelle Abfrage des Bewegungsstatus

Abbildung 3.8: Manuelle Abfrage des Hydrations- und Bewegungsstatus

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse der implementierten Trinkgestenerkennung vorgestellt.

4.1 Modellwahl

Die Wahl der Hyperparameter des MS-TCN-Modells hatte keinen großen Einfluss auf die Performance der Trinkgestenerkennung und wurde daher größtenteils aus dem Paper von Wang et al. übernommen, bis auf drei Unterschiede: (1) die Anzahl der Stages, vier statt zwei, (2) die Anzahl der Layers, zehn statt 9, und (3) die Anzahl der Filter, 64 statt 128 [21]. Diese Anpassungen haben in diesem Anwendungsfall zu einer leicht verbesserten Performance geführt.

Um mehr Trainingsbeispiele zu generieren, wurde eine Segmentierung des Trainingsdatensatzes vorgenommen. Für jede Kombination aus Sequenzlänge (256, 512, 1024, 2048) und Überlappungsrate (0%, 12,5%, 25%, 50%, 75%, 87,5%, 93,75%) wurde der Trainingsdatensatz entsprechend in überlappende Zeitfenster geteilt und darauf jeweils zehn MS-TCN-Modelle mit identischer Architektur trainiert. Dadurch sind 280 Modelle entstanden.

Die Sequenzlänge definiert, wie viele aufeinanderfolgende Frames für die jeweiligen Trainingsfenster zur Eingabe für das Modell verwendet werden.

Die Daten liegen nach der Datenvorbereitung in 16 Hz vor, somit entsprechen die Sequenzlängen folgenden Zeitdauern:

- 256 Frames = 16 Sekunden
- 512 Frames = 32 Sekunden
- 1024 Frames = 64 Sekunden

- 2048 Frames = 128 Sekunden

Die Überlappungsraten definieren, wie stark sich diese Trainingsfenster überschneiden.

Eine Trinkgeste dauert meist zwischen 4 und 12 Sekunden, je nach Trinkgeschwindigkeit und ob ein oder mehrere Schlücke getrunken werden. Dementsprechend sollte die Sequenzlänge mindestens lang genug sein, um komplette Trinkgesten zu erfassen.

Abbildung 4.1 zeigt die Ergebnisse der verschiedenen Sequenzlängen/Überlappungs-Konfigurationen auf dem Evaluationsdatensatz.

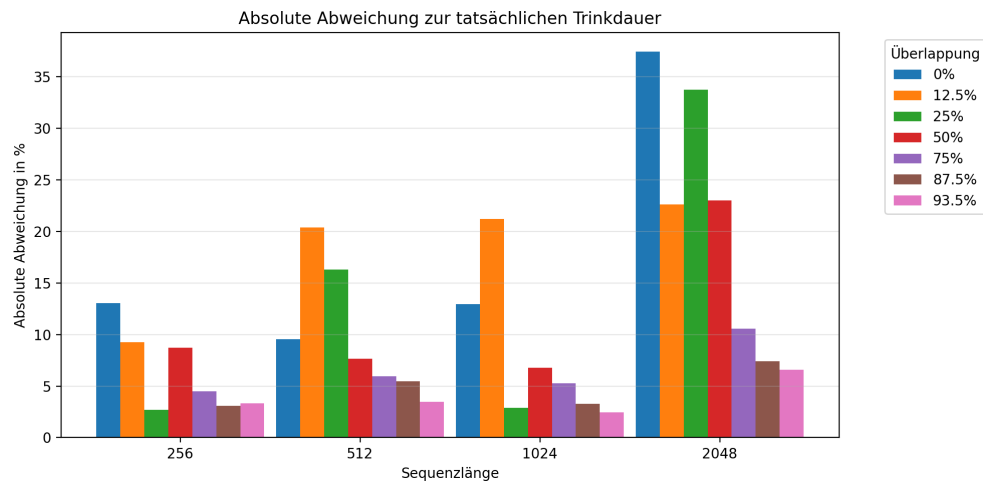


Abbildung 4.1: Durchschnittliche absolute Abweichung aller Modelle trainiert mit verschiedener Segmentierung der Trainingsdaten.

Ziel ist es, die Konfiguration zu finden, die am nächsten an die tatsächliche Trinkdauer herankommt und dabei eine möglichst kleine Varianz über alle Trainingsdurchläufe aufweist. So wird die Konfiguration gewählt, deren Ergebnisse nicht nur zufällig gut sind, sondern konsistent gute Ergebnisse liefern.

Die Ergebnisse in dieser Arbeit werden durch die durchschnittliche absolute Abweichung in % zwischen der vorhergesagten Trinkdauer und der tatsächlichen Trinkdauer, dem F1-Score sowie durch Precision und Recall gemessen.

Der F1-Score setzt sich aus Precision und Recall zusammen und gibt an, wie gut und genau das Modell die Trinkgesten erkennt.

Precision gibt an, wie viele der als Trinkgesten erkannten Ereignisse tatsächlich echte Trinkgesten sind.

$$Precision = TruePositives / (TruePositives + FalsePositives) \quad (4.1)$$

Recall gibt an, wie viele der echten Trinkgesten vom Modell erkannt wurden.

$$Recall = TruePositives / (TruePositives + FalseNegatives) \quad (4.2)$$

Um den F1-Score zu berechnen, wird die Formel

$$F1 - Score = (2 * Precision * Recall) / (Precision + Recall) \quad (4.3)$$

verwendet.

Ohne die Überlappungsraten zu berücksichtigen, ist zu erkennen, dass Sequenzlängen, die näher an der Dauer einer kompletten Trinkgeste liegen (256, 512), tendenziell bessere Ergebnisse erzielen als längere Sequenzlängen (1024, 2048).

Hohe Überlappungsraten (>50%) performen deutlich besser als niedrigere Überlappungsraten. Mit Ausnahme der Überlappungsrate 25% auf den Sequenzlängen 256 und 1024.

Im Folgenden wird auf die Modelle mit unter 5% absoluter Abweichung genauer eingegangen.

In den segmentalen Metriken haben alle Modelle sehr gut abgeschnitten. Fast jedes Modell hat alle echten Trinkgesten konsistent erkannt. Nur bei den False Positives gab es erkennbare Unterschiede.

Die bedeutsameren Metriken für das entwickelte System sind allerdings die frame-wise Metriken, da diese die Trinkdauer genauer abbilden. In den segmentalen Metriken wird nur gezählt, ob eine Trinkgeste erkannt wurde oder nicht. Die frame-wise Metriken geben Aufschluss darüber, wie viele Frames jeder Trinkgeste korrekt erkannt wurden.

Hier sind deutlichere Unterschiede zwischen den Konfigurationen erkennbar. Bei der Sequenzlänge 256 mit den Überlappungsraten 75%, 87,5% und 93,75% ist eine große Varianz zwischen den einzelnen Durchläufen erkennbar. Die Differenz zwischen den besten und schlechtesten Durchläufen beträgt für diese Konfigurationen jeweils 0.0907, 0.0659 und

0.0539. Sie sind somit nicht stabil genug und damit nicht geeignet für den Einsatz in diesem System.

Bei der Sequenzlänge 256 hat die Überlappungsrate 25% am besten abgeschnitten. Mit einem F1-Score von 0.9583 und einer Differenz zwischen dem besten und schlechtesten Durchgang von 0.0169, wäre diese Konfiguration tendenziell geeignet für den Einsatz in der Trinkgestenerkennung.

Das beste Ergebnis hat allerdings die Sequenzlänge 1024 mit der Überlappungsrate 93,75% erzielt. Mit einem durchschnittlichen F1-Score von 0.9688, einem minimalen F1-Score von 0.9443 in den frame-wise Metriken und einer absoluten Abweichung von 2,45% hat diese Konfiguration am besten abgeschnitten. Auch auf den True Positives wies das Modell eine absolute Abweichung von nur 1,01% auf. Außerdem hat es mit 433,3 FPs die wenigsten False Positives erkannt.

Aufgrund dieser Ergebnisse wurde diese Konfiguration für die Segmentierung der Trainingsdaten im Modelltraining gewählt.

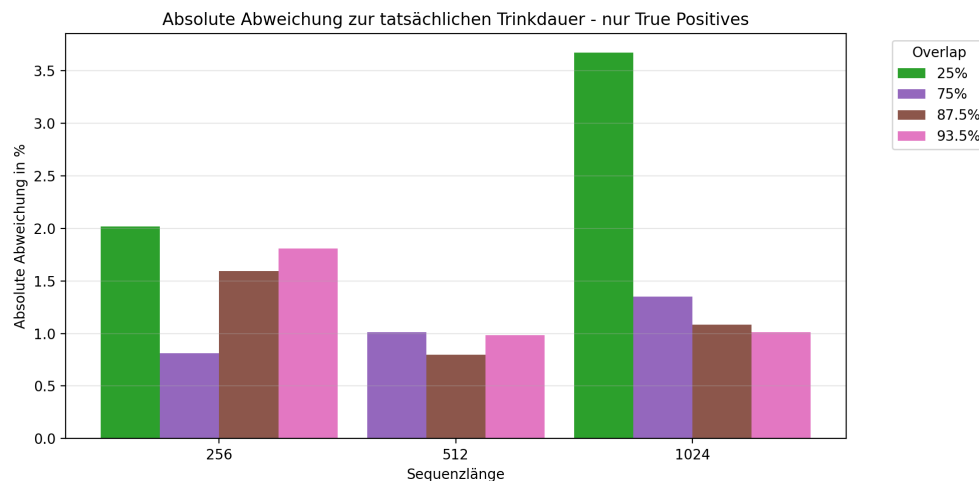


Abbildung 4.2: Durchschnittliche absolute Abweichungen der besten Kombinationen aus Abbildung 4.1 auf den True Positives.

Außerdem wurde die K-Fold Cross-Validation für das Modelltraining getestet. Bei der K-Fold Cross-Validation wird der gesamte Trainingsdatensatz in K gleich große Teile (Folds) aufgeteilt. In diesem Fall mit vier Trainingssessions wird K=4 gesetzt. Der gesamte Datensatz (alle vier Trainingssessions zusammen) wird somit in vier gleich große

Tabelle 4.1: Durchschnittliche Abweichungen der besten Modelle

Modell	erkannte Trinkdauer	Abweichung	erkannte Trinkdauer nur TP	Abweichung nur TP
Seq256 Overlap25%	689.9s	+2.45%	668.5s	-0.73%
Seq256 Overlap75%	700.3s	+3.99%	671.0s	-0.37%
Seq256 Overlap87,5%	685.7s	+1.82%	662.9s	-1.55%
Seq256 Overlap93,5%	687.5s	+2.08%	661.2s	-1.82%
Seq512 Overlap93,5%	693.1s	+2.91%	666.8s	-1.00%
Seq1024 Overlap25%	690.7s	+2.54%	648.7s	-3.68%
Seq1024 Overlap87,5%	692.2s	+2.79%	666.1s	-1.07%
Seq1024 Overlap93,5%	683.4s	+1.47%	667.3s	-0.90%

Tabelle 4.2: Durchschnittliche segmentale Metriken der besten Kombinationen aus Sequenzlängen und Überlappungsraten über 10 Trainingsläufe

Modell	F1-Score	Precision	Recall	TP	FP	FN
Seq256 Overlap25%	0.9815	0.9637	1.0000	98.0	3.7	0.0
Seq256 Overlap75%	0.9828	0.9668	1.0000	98.0	3.5	0.0
Seq256 Overlap87,5%	0.9881	0.9768	1.0000	98.0	2.4	0.0
Seq256 Overlap93,5%	0.9846	0.9699	1.0000	98.0	3.1	0.0
Seq512 Overlap93,5%	0.9840	0.9697	0.9990	97.9	3.1	0.1
Seq1024 Overlap25%	0.9648	0.9323	1.0000	98.0	7.2	0.0
Seq1024 Overlap87,5%	0.9845	0.9698	1.0000	98.0	3.1	0.0
Seq1024 Overlap93,5%	0.9875	0.9775	0.9980	97.8	2.3	0.2

Tabelle 4.3: Durchschnittliche frame-wise Metriken der besten Kombinationen aus Sequenzlängen und Überlappungsraten über 10 Trainingsläufe

Modell	F1-Score	Precision	Recall	TP	FP	FN
Seq256 Overlap25%	0.9583	0.9500	0.9671	10482.1	556.4	356.9
Seq256 Overlap75%	0.9577	0.9441	0.9732	10548.1	657.2	290.9
Seq256 Overlap87,5%	0.9621	0.9579	0.9675	10486.2	485.4	352.8
Seq256 Overlap93,5%	0.9606	0.9547	0.9674	10485.3	514.4	353.7
Seq512 Overlap93,5%	0.9600	0.9499	0.9708	10522.8	566.8	316.2
Seq1024 Overlap25%	0.9413	0.9329	0.9503	10300.2	750.4	538.8
Seq1024 Overlap87,5%	0.9587	0.9495	0.9688	10500.3	575.6	338.7
Seq1024 Overlap93,5%	0.9647	0.9609	0.9688	10500.6	433.3	338.4

Tabelle 4.4: Weitere frame-wise Metriken

Modell	Bester F1-Score	Schlechtester F1-Score	Differenz
Seq256 Overlap25%	0.9648	0.9479	0.0169
Seq256 Overlap75%	0.9794	0.8887	0.0907
Seq256 Overlap87,5%	0.9783	0.9124	0.0659
Seq256 Overlap93,5%	0.9806	0.9267	0.0539
Seq512 Overlap93,5%	0.9774	0.9338	0.0436
Seq1024 Overlap25%	0.9520	0.9135	0.0385
Seq1024 Overlap87,5%	0.9746	0.9158	0.0588
Seq1024 Overlap93,5%	0.9765	0.9443	0.0322

Teile aufgeteilt. Für jeden Fold wird einer dieser vier Teile als Validierungsmenge verwendet, während die restlichen K-1 Teile zum Training des Modells genutzt werden. Dies wird für alle K Folds wiederholt, wodurch K verschiedene Modelle entstehen.

Die Performance der Modelle, die mit der K-Fold Cross-Validation trainiert wurden, hat stark variiert. Dadurch, dass der bereits kleine Trainingsdatensatz noch weiter aufgeteilt wird, entstehen sehr kleine Trainingsmengen für jedes Modell. Manche Trainingssessions enthalten so möglicherweise gar keine oder nur sehr wenige Trinkgesten. Außerdem ist nicht klar, welches der K-Modelle später für die Analyse der Tagesdaten verwendet werden soll, da die einzige Metrik zum Vergleichen der Modelle der F1-Score auf dem kleinen Validierungsdatensatz ist. Hier hatten alle K-Modelle sehr ähnliche F1-Scores und das Modell mit dem besten F1-Score hat nicht zwangsläufig die besten Ergebnisse auf dem Evaluationsdatensatz erzielt. Deshalb wurde sich dagegen entschieden die K-Fold Cross-Validation zu verwenden. Stattdessen wurde ein einzelnes Modell auf dem gesamten Datensatz trainiert.

4.2 Trainingssession-Versuche

Im Laufe der Entwicklung wurden verschiedene Methoden ausprobiert, um die Hürde für das Einrichten der Smartwatch-Applikation so gering wie möglich zu halten und dabei eine gute Erkennungsrate im Alltag zu erzielen. Angefangen wurde mit der direkten Aufnahme von je drei Trinkgesten pro Kombination aus Position (stehend, sitzend, liegend) und Behältertyp (Glas, Tasse, Flasche). Da hier nur positive Trainingsbeispiele aufgenommen wurden, hat das Modell auf echten Alltagsdaten schlecht abgeschnitten.

Der nächste Versuch war die Aufnahme von vier 1-Stunden-Trainingssessions mit je einer Trinkgeste alle 5 Minuten. Das mit diesem Datensatz trainierte Modell erreichte einen F1-Score von 0.85 auf dem Evaluationsdatensatz bereits vor dem Finden der optimalen Konfiguration mit einer Sequenzlänge von 2048 und 0% Überlappung der Trainingsfenster.

Beim nächsten Versuch wurde eine 1-Stunden-Trainingssession mit einer Trinkgeste alle 2-3 Minuten durchgeführt, in der alle 15 Minuten der Behälter gewechselt wurde (kleine und große Tasse, Glas, Flasche). Dies hat mit einem F1-Score von etwa 0.7 auf dem Evaluationsdatensatz nicht zufriedenstellend genug funktioniert.

Schließlich wurde die aktuell verwendete Methode mit vier 30-Minuten-Trainingssessions mit einer Trinkgeste alle 2-3 Minuten eingeführt. Diese Methode hat sich als guter Kompromiss zwischen Trainingsaufwand und Erkennungsleistung erwiesen.

4.2.1 Finetuning

Es wurde zudem probiert, ein Modell mit den Datensätzen des Papers von Wang et al. vorzutrainieren und dann mit eigenen Trainingssessions zu finetunen. Dies hat nicht gut funktioniert, da die Datensätze des Papers mit der Shimmer3 IMU aufgenommen wurden und dieser im Vergleich zur Smartwatch andere Sensorkalibrierungen und Genauigkeiten hat. Das vortrainierte Modell konnte die Bewegungsmuster der Smartwatch-Daten nicht gut erkennen und erreichte nur einen F1-Score von etwa 0.3 auf dem Evaluationsdatensatz. Das finetuned Modell hat sich nicht signifikant von dem vortrainierten Modell unterschieden.

4.3 Einschränkungen

Unter bestimmten Umständen kann das System nicht korrekt funktionieren.

Es werden nur Trinkgesten erkannt, die mit der Hand ausgeführt werden, an der die Smartwatch getragen wird. Trinken mit beiden Händen wird noch erkannt, wenn dies in den Trainingssessions beachtet wurde. Wird mit der Hand getrunken, an der die Smartwatch nicht getragen wird, kann die Trinkgeste nicht erkannt werden.

Die Vielfalt individueller Trinkgewohnheiten macht es schwierig, ein Modell zu trainieren, das alle Verhaltensweisen zuverlässig abbildet. Insbesondere, wenn die Schwelle für das Einrichten so klein wie möglich gehalten werden soll. Um alle individuellen Trinkgewohnheiten abzudecken, sind umfangreichere Trainingsdaten erforderlich, was den Einrichtungsprozess für die Nutzer erschweren würde.

In eigenen Tests wurde festgestellt, dass das Trinktempo oft von der Temperatur des Getränks abhängt. Sehr kalte oder sehr heiße Getränke werden langsamer getrunken. Wenn die Trainingsdaten dies nicht ausreichend repräsentieren, kann dies die Erkennungsleistung im Alltag in Bezug auf die Trinkdauer beeinträchtigen. Trinkt ein Nutzer an einem Tag ausschließlich heißen Tee, dann wird dies die Trinkdauer stark erhöhen und das System damit die tatsächlich getrunkene Menge überschätzen.

Trainingssessions können zudem verzerrte Bewegungsmuster liefern. Nutzer führen Gesten während eines Trainings häufig bewusst und konzentriert aus. Trinkbewegungen im Alltag sind dagegen oft unbewusst und stärker variabel.

Auch der Durst beeinflusst das Trinkverhalten. So tendieren stark durstige Personen dazu, schneller zu trinken, sodass die Dauer einer Trinkgeste im Alltag von den im Training erfassten Mustern abweichen kann.

Solche Abweichungen können die Erkennungsleistung in realen Einsatzszenarien reduzieren. Gegebenenfalls sind zusätzliche Trainingsdaten erforderlich, um die Robustheit der Erkennung zu erhöhen.

Sporadisches Tragen, Trinken mit der falschen Hand oder schlechte Trainingsdaten können von dem System nicht ausgeglichen werden. Die Selbstüberwachung geschieht in Eigenverantwortung des Nutzers.

4.3.1 Akkulaufzeit

Die kontinuierliche Erfassung von hochfrequenten Sensordaten auf der Smartwatch führt zu einem erhöhten Energieverbrauch, wodurch die Akkulaufzeit der Smartwatch stark beeinträchtigt wird. Die durchschnittliche Akkulaufzeit einer Pixelwatch 1 wird mit etwa 24 Stunden bei normaler Nutzung angegeben [5]. In Tests wurde festgestellt, dass die Smartwatch bei aktiver Datenerfassung und stündlicher Datenübertragung eine Akkulaufzeit von etwa 10 Stunden aufweist. Dadurch muss die Smartwatch in der Regel einmal täglich

zwischendurch aufgeladen werden. Aber schon eine Ladung für 15-30 Minuten reicht aus, um die Smartwatch den restlichen Tag über tragen zu können.

Neuere Smartwatch-Modelle mit größeren Akkus und effizienteren Prozessoren könnten die Akkulaufzeit weiter verbessern.

Um die Laufzeit zu maximieren, wurden verschiedene Optimierungsmaßnahmen implementiert. Dazu zählen die stündliche, gebündelte und komprimierte Datenübertragung anstatt kontinuierlichem Streaming sowie das Pausieren der Datensammlung, wenn die Smartwatch nicht getragen wird.

5 Fazit und Ausblick

Diese Arbeit zeigt, dass die Erkennung von Trinkgesten und die Analyse des Hydrations- und Bewegungsstatus mit handelsüblichen Smartwatches möglich ist. Die durchgeführte Evaluation zeigt, dass Modelle, die auf individuell aufgenommenen Trainingssessions basieren, eine hohe Genauigkeit bei der Erkennung von Trinkgesten im Alltag des Nutzers erreichen können. Den größten Einfluss auf die Erkennungsleistung haben dabei die Wahl der Sequenzlänge und Überlappungsrate bei der Segmentierung der Trainingsdaten. Hohe Überlappungsraten performen deutlich besser als niedrige Raten. Das liegt daran, dass durch die hohe Überlappung, aus dem kleinen Trainingsdatensatz mehr Trainingsbeispiele generiert werden können.

Die Wahl der Sequenzlänge hat ebenfalls einen Einfluss auf die Erkennungsleistung. So performen niedrige Sequenzlängen besser als hohe Sequenzlängen. Aus den Ergebnissen der Evaluation lässt sich schließen, dass in niedrigen Sequenzlängen das Verhältnis zwischen Trinkgesten und Nicht-Trinkgesten für einzelne Trainingsfenster ausgeglichener ist und damit die Erkennung begünstigt. Ist die Sequenzlänge allerdings zu niedrig, kann es keine ganzen Trinkgesten mehr erfassen. Sobald niedrige Sequenzlängen mit hohen Überlappungsraten kombiniert werden, steigt die Varianz der Modellergebnisse. Das bedeutet, dass die Ergebnisse bei gleicher Konfiguration in verschiedenen Trainingsdurchläufen stark variieren können.

Das beste Modell wurde mit einer Segmentierung der Trainingsdaten in Sequenzen der Länge 1024 und einer Überlappungsrate von 93,75% trainiert. Dieses Modell erreichte auf dem Evaluationsdatensatz einen durchschnittlichen F1-Score von 0.9647 in den frame-wise Metriken und eine durchschnittliche absolute Abweichung von 2,45% gegenüber der tatsächlichen Trinkdauer. Der am schlechtesten performende Trainingslauf des besten Modells erreichte immer noch einen F1-Score von 0.9443 auf den frame-wise Metriken.

Senioren oder deren Betreuer können durch die Nutzung des Assistenzsystems frühzeitig auf eine unzureichende Flüssigkeitszufuhr oder einen Mangel an körperlicher Aktivität

hingewiesen werden. Dadurch könnten sie länger selbstständig im eigenen Zuhause leben und ihre Lebensqualität erhalten.

Für die Zukunft ist ein generalisierendes Modell wünschenswert, welches ohne individuelle Trainingssessions auskommt. Dieses könnte durch die Sammlung von Trainingsdaten vieler Nutzer und vieler verschiedener Smartwatches denkbar sein. Damit würde die Hürde für die Nutzung deutlich gesenkt und die Akzeptanz des Systems erhöht werden.

Literaturverzeichnis

- [1] BLACKMAN, Stephanie ; MATLO, Claudine ; BOBROVITSKIY, Charisse ; WALDOCH, Ashley ; FANG, Mei L. ; JACKSON, Piper ; MIHAILIDIS, Alex ; NYGÅRD, Louise ; ASTELL, Arlene ; SIXSMITH, Andrew: Ambient Assisted Living Technologies for aging well: A scoping review. In: *Journal of Intelligent Systems* 25 (2015), Mar, Nr. 1, S. 55–69
- [2] BULL, Fiona C. ; AL-ANSARI, Salih S. ; BIDDLE, Stuart ; BORODULIN, Katja ; BUMAN, Matthew P. ; CARDON, Greet ; CARTY, Catherine ; CHAPUT, Jean-Philippe ; CHASTIN, Sebastien ; CHOU, Roger ; AL. et: World Health Organization 2020 guidelines on physical activity and sedentary behaviour. In: *British Journal of Sports Medicine* 54 (2020), Nov, Nr. 24, S. 1451–1462
- [3] FARHA, Yazan A. ; GALL, Jürgen: MS-TCN: Multi-stage temporal convolutional network for action segmentation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), Jun, S. 3570–3579
- [4] FENG, Shan ; MÄNTYMÄKI, Matti ; DHIR, Amandeep ; SALMELA, Hannu: How self-tracking and the quantified self promote health and well-being: Systematic review. In: *Journal of Medical Internet Research* 23 (2021), Sep, Nr. 9
- [5] GOOGLE: *Google Pixel Watch: die erste Smartwatch von Google*. <https://blog.google/intl/de-de/produkte/hardware/google-pixel-watch-erste-smartwatch-von-google/>. oct 2022. – Blogpost, abgerufen am 20.01.2026
- [6] HAMRICK, Irene ; NORTON, Derek ; BIRSTLER, Jen ; CHEN, Guanhua ; CRUZ, Laura ; HANRAHAN, Lawrence: Association between dehydration and Falls. In: *Mayo Clinic Proceedings: Innovations, Quality & Outcomes* 4 (2020), Jun, Nr. 3, S. 259–265
- [7] KENNEY, W. L. ; CHIU, Percy: Influence of age on thirst and fluid intake. In: *Medicine & Science in Sports & Exercise* 33 (2001), Sep, Nr. 9, S. 1524–1532

- [8] KINGA, Diederik ; ADAM, Jimmy B. u. a.: A method for stochastic optimization. In: *International conference on learning representations (ICLR)* Bd. 5 California (Veranst.), 2015
- [9] LIU, Kai-Chun ; HSIEH, Chia-Yeh ; HUANG, Hsiang-Yun ; CHIU, Li-Tzu ; HSU, Steen Jun-Ping ; CHAN, Chia-Tai: Drinking Event Detection and Episode Identification Using 3D-Printed Smart Cup. In: *IEEE Sensors Journal* 20 (2020), Nr. 22, S. 13743–13751
- [10] MASENTO, Natalie A. ; GOLIGHTLY, Mark ; FIELD, David T. ; BUTLER, Laurie T. ; REEKUM, Carien M. van: Effects of hydration status on cognitive performance and mood. In: *British Journal of Nutrition* 111 (2014), Nr. 10, S. 1841–1852
- [11] NOWOSSADECK, S.: Demografischer Wandel, pflegebedürftige und der Künftige Bedarf an Pflegekräften. In: *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz* 56 (2013), Jul, Nr. 8, S. 1040–1047
- [12] REEDER, Blaine ; DAVID, Alexandria: Health at hand: A systematic review of Smart Watch uses for Health and Wellness. In: *Journal of Biomedical Informatics* 63 (2016), Oct, S. 269–276
- [13] SCHMIDT, Andreas T. ; ENGELEN, Bart: The Ethics of Nudging: An Overview. In: *Philosophy Compass* 15 (2020), Feb, Nr. 4
- [14] SCHOLS, J.M.G.A. ; DE GROOT, C.P.G.M. ; VAN DER CAMMEN, T.J.M. ; OLDE RIKKERT, M.G.M.: Preventing and treating dehydration in the elderly during periods of illness and warm weather. In: *The Journal of nutrition, health and aging* 13 (2009), Feb, Nr. 2, S. 150–157
- [15] SOUBAM, Sonia ; AGRAWAL, Manisha ; NAIK, Vinayak: Using an Arduino and a Smartwatch to Measure Liquid Consumed From Any Container. In: *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, IEEE, 2017, S. 464–467
- [16] STATISTISCHES BUNDESAMT (DESTATIS): *Personal in Pflegeeinrichtungen*. <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Pflege/Tabellen/personal-pflegeeinrichtungen.html>. 2024. – Tabelle, abgerufen am 20.01.2026
- [17] STATISTISCHES BUNDESAMT (DESTATIS): *Pflegebedürftige insgesamt*. <https://www.destatis.de/DE/Presse/Pressemitteilungen/Grafiken/Sozi>

- ales/2024/_Interaktiv/20241218-pflege-beduerftige-insgesamt.html?showChartData=1. dec 2024. – abgerufen am 20.01.2026
- [18] TANAKA, Hirofumi ; MONAHAN, Kevin D. ; SEALS, Douglas R.: Age-predicted maximal heart rate revisited. In: *Journal of the American College of Cardiology* 37 (2001), Jan, Nr. 1, S. 153–156
- [19] TIGERDATA: *TigerData Documentation: TimescaleDB*. <https://www.tigerdata.com/docs/#TimescaleDB>. 2026. – abgerufen am 21.01.2026
- [20] VOLKERT, Dorothee ; BECK, Anne M. ; CEDERHOLM, Tommy ; CRUZ-JENTOFT, Alfonso ; GOISSER, Sabine ; HOOPER, Lee ; KIESSWETTER, Eva ; MAGGIO, Marcello ; RAYNAUD-SIMON, Agathe ; SIEBER, Cornel C. ; AL. et: Espen guideline on Clinical Nutrition and hydration in Geriatrics. In: *Clinical Nutrition* 38 (2019), Feb, Nr. 1, S. 10–47
- [21] WANG, Chunzhuo ; KUMAR, T. S. ; DE RAEDT, Walter ; CAMPS, Guido ; HALLEZ, Hans ; VANRUMSTE, Bart: Drinking gesture detection using wrist-worn imu sensors with multi-stage temporal convolutional network in free-living environments. In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2022), Jul, S. 1778–1782
- [22] ZHUANG, Wei ; CHEN, Yi ; SU, Jian ; WANG, Baowei ; GAO, Chunming: Design of human activity recognition algorithms based on a single wearable IMU sensor. In: *International Journal of Sensor Networks* 30 (2019), Nr. 3, S. 193

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original