Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

George-Alexandru Negru

Ganganalyse als Authentifizierungsverfahren mittels Smartphone-Sensorik

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

George-Alexandru Negru

Ganganalyse als Authentifizierungsverfahren mittels Smartphone-Sensorik

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Robert Fitz
Zweitgutachter : Prof. Dr. -Ing. Mark Hensel

Eingereicht am: 03. Mai 2019

George-Alexandru Negru

Thema der Bachelorthesis

Ganganalyse als Authentifizierungsverfahren mittels Smartphone-Sensorik

Stichworte
Ganganalyse, Sensorik, Authentifizierung, Biometrie, Maschinelles Lernen

Kurzzusammenfassung

Diese Arbeit umfasst die Modellierung und anschließende Implementierung eines biometrischen Authentifizierungssystems für Smartphones. Die Authentisierung findet mittels beschleunigungsbasierter biometrischer Gangerkennung statt. Die Modellierung wird mit Hilfe der Software Matlab realisiert und beruht auf dem Konzept des maschinellen Lernens. Für die Konzeptrealisierung werden biometrische Daten von zehn Kandidaten untersucht und ausgewertet. Das fertige Modell wird anschließend als Android-Applikation mit Hilfe von Java und der Android Studio-Software implementiert.

George-Alexandru Negru

Title of the paper

Gait analysis as a mean of Authentication using mobile Sensors

Keywords

Gait analysis, Sensors, Authentication, Biometrics, Machine Learning

Abstract

This work comprises primarly of the simulation and implementation of a biometric authentication system for smartphone devices. The authentication takes place via accelerometer-based biometric gait recognition. The modeling of the System is realized with help of the software Matlab and rests on the concept of machine learning. To help create this model, biometric data from ten candidates will be examined and evaluated. The finished model is then implemented on an Android Smartphone device with the help of Java and Android Studio Software.

Danksagungen

Zuerst möchte ich mich ganz herzlich bei Prof. Dr.-Ing. Robert Fitz bedanken. Er hat mir diese Arbeit überhaupt erst ermöglicht. Außerdem gab er mir zahlreiche Anregungen und betreute mich sehr kompetent während der Erstellung dieser Arbeit. Weiterhin möchte ich mich auch bei meinem Zweitgutachter Prof. Dr. -Ing. Mark Hensel bedanken.

An letzter Stelle möchte ich mich bei meiner Freundin Theresa, meiner Tochter Alisia, meinen Eltern und meinen Schwiegereltern für die tatkräftige Unterstützung bedanken, die weit über die Erstellung der Arbeit hinausgeht.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

`.csv` Comma-Separated Values.

ACC Accuracy (dt. Genauigkeit).

APK Android Package Kit (dt. Android-Programmpakete).

APP Application (dt. Applikation).

ARFF Attribute-Relation File Format.

ARM Advanced RISC Machines.

ASCII American Standard Code for Information Interchange.

bzw. beziehungsweise.

ca. circa.

CLI Command Line Interface (dt. Kommandozeileninterpreter).

DFT Diskrete Fourier-Transformation.

EER Equal Error Rate (dt. Gleichfehlerrate).

FAR False Acceptance Rate (dt. Falschakzeptanzrate).

FRR False Rejection Rate (dt. Falschrückweisungsrate).

GPS Global Positioning System (dt. Globales Positionsbestimmungssystem).

GUI Graphical User Interface (dt. Grafische Benutzeroberfläche).

i.d.R. in der Regel.

IBAN International Bank Account Number.

IDE Integrated Development Environment (dt. Integrierte Entwicklungsumgebung).

k-NN k-nächste-Nachbarn (en. k-Nearest Neighbor).

MEMS Microelectromechanical systems.

NFC Near Field Communication (dt. Nahfeldkommunikation).

OTP One Time Passwort.

PIN Personal Identification Number(dt. Persönliche Geheimnummer).

PUK Personal Unblocking Key.

RAM Random-Access Memory.

RMS Root Mean Square.

ROC Receiver operating characteristic (dt. Grenzwertoptimierungskurve).

SVM Support Vector Machines (dt. Stützvektormaschine).

TAN Transaction Number.

TCP/IP Transmission Control Protocol/Internet Protocol.

USB Universal Serial Bus.

usw. und so weiter.

VM Virtuelle Maschine.

WEKA Waikato Environment for Knowledge Analysis.

WLAN Wireless Local Area Network (dt. drahtloses lokales Netzwerk).

WWW World Wide Web (dt. Weltweites Netz).

z. B. zum Beispiel.

Symbolverzeichnis

γ Gamma.

μ Mittelwert.

Φ Kernel Hilfsfunktion.

λ_C Schwerpunktwellenlänge.

σ Standardabweichung.

ξ Minimaler Abstand eines Datenpunktes zur Hyperebene.

1 Einführung

1.1 Motivation

Mit der stetigen Entwicklung der Technik in den letzten Jahren sind wir im Stande immer kleinere und stärkere elektronische Bauteile zu entwickeln. Somit können ganz schnelle Prozessoren, Speicherbausteine unterschiedlicher Sensoren und andere elektronische Bauteile zusammen auf eine sehr kleine Platine zusammengebracht werden. Das Ganze führt dazu, dass immer kleinere und leistungsfähigere elektronische Geräte produziert werden. Heutzutage ist ein großer Anteil der Menschheit im Besitz von solch elektronischen Geräten.

Die konventionellen Gegenstände, die wir einmal gekannt haben, werden immer weniger. Somit sind normale Uhren, die nur benutzt werden um die Uhrzeit abzulesen, Brillen die nur für die Sichtverbesserung eingesetzt sind oder Telefone, die nur zum Telefonieren benutzt werden, ersetzt von sogenannten Smart-Geräten. Diese neue Art von Geräten kann jetzt viel mehr. Eine Smart-Uhr kann heutzutage nicht nur die Uhrzeit anzeigen, sie dient gleichzeitig auch als Schrittzähler, Herzschrittmesser und wird außerdem zum Telefonieren benutzt. Das sind nur einige der vielen Eigenschaften, die sich aktuell rasant entwickeln. Von allen Smart-Geräten ist das Smartphone das am weitesten Verbreitete. Mit Entwicklung von stetig schnellerem Internet für unterwegs wie zum Beispiel (z. B.) 4G und jetzt sogar das 5G Netz, ist es sehr einfach das Smartphone als Brücke zur digitalen Welt zu nutzen.

Laut dem aktuellen Zenith Mobile Advertising Forecast [25] besitzen im Jahr 2018 weltweit (in 52 Märkten) 66 Prozent der Menschen ein Smartphone. Die Verwaltung von E-Mails ist die Internetaktivität Nummer eins, gefolgt von der Nutzung der Suchmaschinen und dem Online-Shopping. Weitere Aktivitäten, die unter Smartphonebenutzern beliebt sind, ist die Benutzung von sozialen Netzwerken z. B. Facebook, Instagram oder Twitter, das Streamen von Bewegtbildern, die Nutzung von Cloud-Diensten, Instant Messaging,

Online-News lesen und vieles mehr. Eine weitere Funktion, die von Smartphonebenutzern immer häufiger verwendet wird, ist das bargeldlose Zahlen mit einem Smart-Gerät.

Die Datenmengen an Informationen, die sich heutzutage auf einem Smartphone befinden, sind sehr groß und variieren von persönlichen Bildern und E-Mails bis zu Login-Daten für diverse Anbieter. Ebenso befinden sich Bankdaten für die mobile Bezahlung auf dem Smartphone, da viele Banken diese digitale Zahlungsdienstleistung anbieten. Um vor Missbrauch zu schützen gibt es sehr viele Authentifizierungsverfahren, die von einfachen vierstelligen Personal Identification Number(dt. Persönliche Geheimnummer) (PIN), einem Passwort oder einem bestimmten Muster bis hin zu biometrischen Verfahren wie Fingerabdruck- oder einem Iris-Scanner reichen. Für eine elektronische Banküberweisung [22, S.1-3] wird zum Beispiel eine Multi-Faktor-Authentifizierung verwendet, es wird das International Bank Account Number (IBAN), ein geheimes Passwort und eine einmalige verwendbare Transaktionsnummer Transaction Number (TAN) benötigt.

Die offene Frage bleibt wie sicher diese Verfahren gegen Missbrauch sind und inwiefern die privaten Daten der Nutzer in Wirklichkeit geschützt werden. Die Anmeldung am Smartphone erfolgt meistens mit der Eingabe einer vier- bis sechs-stelligen PIN oder eines Standardpasswortes, welches in der Regel (i.d.R.) aus sechs Zeichen oder aus einem einfachen Muster besteht.

Mit zunehmender Anzahl an Geräten in Verbindung mit nicht-ausgereifter Sicherheit wird diese Technologie für kriminelle Machenschaften immer interessanter. Solche Authentifizierungsverfahren können leicht von Angreifern ausgenutzt werden. Durch einfache Beobachtung oder durch Techniken wie der Brute-Force-Methode, bei der alle möglichen Kombinationen ausprobiert werden, kann so in kurzer Zeit das richtige Passwort ermittelt werden. Auch komplizierte PINs, Passwörter mit Sonderzeichen oder komplexere Muster können durch mehrfache Beobachtung von Angreifern gemerkt werden (diese Methode ist als Shoulder Surfing bekannt). Gegen die Brute-Force-Methode ist als Schutz eine begrenzte Anzahl an erlaubten Fehlversuchen eingeführt worden. I.d.r. sind dies drei Versuche, danach ist die ursprüngliche PIN, das Muster oder das Passwort nicht mehr gültig, sodass eine Neuanmeldung nur mit einem Master-Passwort oder Personal Unblocking Key (PUK)(Super PIN) erfolgen kann.

Viele Firmen erhöhen die Sicherheit ihrer Smartphones, indem sie ein zusätzliches Gerät (Token) mit einem One Time Passwort (OTP)-Generator ihren Mitarbeitern geben. Die Sicherheit wird zwar erhöht, aber das zusätzliche Gerät muss überall mitgeführt werden. Bei Verlust besteht für den User keine Möglichkeit mehr sich anzumelden. Die große

Problematik stellt sich bei diesem Verfahren, dass der Angreifer das Passwort beliebig oft nutzen kann, sobald er dies zuvor einmal entziffert hat.

Um eine bessere Sicherheit gewährleisten zu können und speziell auf einen bestimmten Nutzer zuzuschneiden, gibt es als Alternative die biometrische Authentifizierung. Der Begriff Biometrie ist aus dem Griechischen abgeleitet und setzt sich aus *bios*(Leben) und *metron* (Maß) zusammen [9, S.13-15]. Die Biometrie ist somit nichts anderes als die Nutzung der einzigartigen persönlichen Merkmale des Nutzers zur Verifikation oder Identifikation. Die meist verbreiteten biometrischen Authentifizierungsverfahren nutzen Merkmale wie Fingerabdruck, die Regenbogenhaut- (Iris-), Netzhaut- (Retina-) oder Stimmenmuster als Vergleich für den Authentifizierungsprozess.

Ein Vorteil der Biometrie ist, dass der Benutzer sich keine komplizierten Muster oder Passwörter merken muss. Er hat sein Passwort immer dabei und kann dies auch nicht verlieren. Als Nachteil sollte man die Erkennungsgenauigkeit beachten, daher wird i.d.R. zudem noch ein Passwort oder PIN eingeführt, welches den Vorteil hat, eine hundertprozentige Genauigkeit zu haben. Dieses bietet die Möglichkeit, entweder die eine Methode oder die andere Methode zu nutzen und ist als ein-Faktor-Authentifizierung bekannt.

Folgende Frage ergibt sich daraus: Was nützt solch ein enorm sicheres Authentifizierungsverfahren , wenn es ganz leicht umgegangen werden kann? Wie wird das nicht-gesperrte Gerät bei einem Verlust oder Diebstahl geschützt? An dieser Stelle wird das Konzept von Multi-Faktor-Authentifizierung eingeführt.

Die Multi-Faktor-Authentifizierung führt mindestens noch eine weitere Sicherheitsschicht hinzu. Der Vorteil neben des wesentlich erhöhten Sicherheitsfaktors ist, dass der Vorgang passiv erfolgt und der Nutzer keine weitere Verantwortung trägt. Ein Beispiel wäre die Identifikation anhand der Analyse des Tippverhaltens. Dieser Prozess erfolgt dann im Hintergrund und wird nur in bestimmten Situationen durchgeführt. Dies ist der Fall, wenn der User die Tastatur benutzt um etwas einzutippen.

Diese Arbeit beschäftigt sich mit einem ähnlichen Verfahren, es wird versucht die Identifikation oder Verifikation der Identität einer Person anhand der Ganganalyse durchzuführen.

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist das oben genannte Authentifizierungsverfahren mit Hilfe der Ganganalyse zu untersuchen, ein funktionierendes Konzept zu entwerfen und eine erfolgreiche Implementierung auf das Smartphone zu realisieren.

Das Konzept wird mit Hilfe der Software MATLAB modelliert. Es werden unterschiedliche Methoden untersucht, simuliert und ausgewertet. Nach Auswertung der Ergebnisse wird schließlich das beste Modell rausgesucht. Mit Hilfe der Parameter aus MATLAB erfolgt eine Implementierung auf ein ANDROID Smartphone. Die tatsächliche Programmierung der Software erfolgt in ANDROID STUDIO. Als Programmiersprache wurde JAVA ausgewählt.

In dieser Arbeit wird das Authentifizierungsverfahren nur während der Aktivität Gehen betrachtet. Auch das Gehen findet für jeden Kandidaten unter den gleichen Bedingungen, sowohl bei der Aufnahme als auch bei der Identifizierung, statt. Andere Szenarien werden hier nicht untersucht, eine Aktivitätserkennung wird in der Software nicht eingebaut, da der Arbeitsaufwand und Schwierigkeitsgrad diese Bachelorarbeit überschreiten würden und es zusätzlich für die Ziele dieser Arbeit irrelevant ist.

Um sicherzustellen, dass diese Ziele am Ende erreicht wurden, müssen durch die vorliegende Arbeit folgende Anforderungen eingehalten werden:

- das Konzept soll in der Lage sein die gelieferten Daten einlesen zu können und richtig auszuwerten
- das resultierende Modell soll bei der Klassifizierung von Kandidaten eine Genauigkeit von mindestens 90% aufweisen
- die Auswertungszeit soll klein gehalten werden, es wird eine Zeit von bis zu 1,5 Sekunden ausgewählt
- die Implementierung soll die gleichen Merkmale zeigen wie das Modell, eine Toleranz von ca. 5% bis 10% ist erlaubt
- die Software soll über eine Anzeige verfügen, bei der die Genauigkeit und das Ergebnis der Untersuchung angezeigt werden
- der Benutzer soll in der Lage sein die Software über eine eingebaute Steuerung zu bedienen.

- die Software soll in der Lage sein eine erfolgreiche Identifizierung in den gegebenen Parametern durchzuführen.

1.3 Gliederung

Diese Arbeit ist folgendermaßen gegliedert: die nachfolgende Abbildung 1.1 stellt den strukturellen Aufbau der Arbeit dar.

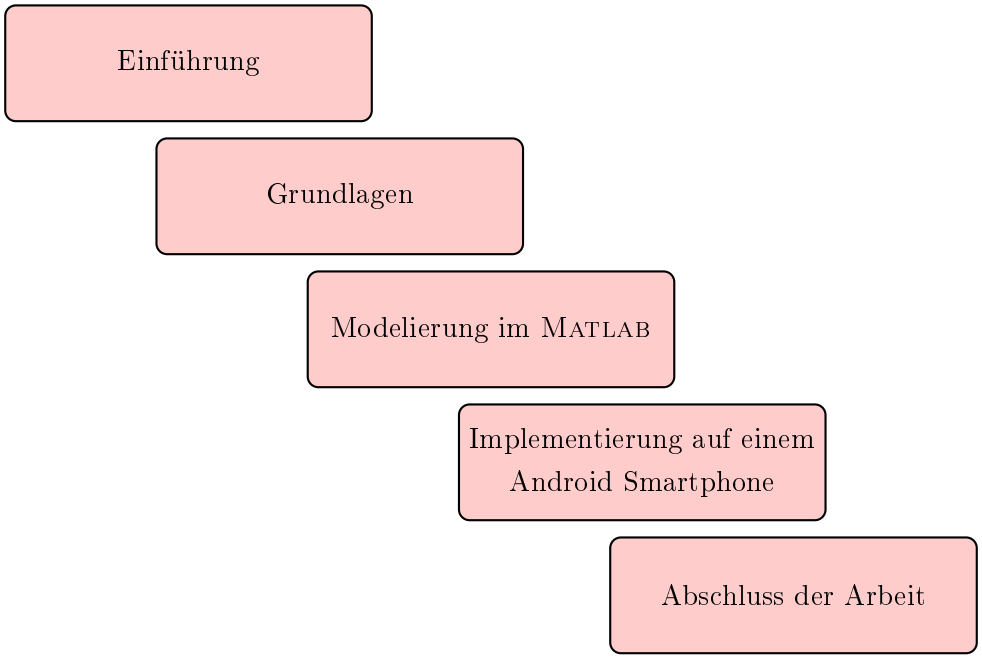

Abbildung 1.1: Struktur der Arbeit

Die Heranführung an das Thema und anschließend ein Überblick zur Vorgehensweise sowie zum Aufbau der Arbeit wurden bereits im Kapitel 1 präsentiert. Als nächstes werden im Kapitel 2 die Grundlagen für diese Thematik präsentiert und definiert. Dazu zählen eine Einführung der verwendeten Hardware und Sensorik und Aspekte aus der Authentifizierung (insbesondere die Ganganalyse als biometrisches Verfahren). Letztendlich wird auf die Software für die Lösungen der Aufgaben eingegangen.

Kapitel 3 zeigt die notwendigen Schritte für ein Authentifizierungsverfahren mittels Ganganalyse anhand eines konkreten Beispiels. Es werden unterschiedliche Methoden und Parameter mit Hilfe der Software MATLAB vorgestellt.

Darauf folgt eine einfache Implementierung des Konzeptes im Kapitel 4. Hier wird mit Hilfe der Erkenntnisse und Parameter aus den vorherigen Kapiteln die technische Umsetzung in Software beschrieben und festgelegt. Eine Auswertung fasst die Ergebnisse anhand der durchgeführten Funktionstests zusammen.

Den Abschluss bildet Kapitel 5 mit einem Fazit und einem Ausblick.

2 Grundlagen

Dieses Kapitel behandelt die notwendigen Grundlagen, die für das Verständnis dieser Arbeit notwendig sind.

2.1 Hardware

2.1.1 Smartphone

Das einzige Stück Hardware, welches für diese Arbeit verwendet wurde, ist das Smartphone. In diesem Abschnitt werden die allgemeinen Funktionen und Eigenschaften des Smartphones präsentiert. Das Smartphone ist ein Mobiltelefon, welches außerhalb der Funktion zum Telefonieren über eine Reihe von weiteren Eigenschaften verfügt, welche normalerweise in einem Computer eingebaut sind z. B. ein Betriebssystem. Die am häufigsten verwendeten Betriebssysteme sind das Android OS oder Apple iOS. Die Programme oder Software, welche auf einem Smartphone laufen, lassen sich als Application (dt. Applikation) (APP) bezeichnen. Der Prozessor übernimmt in einem Smartphone, wie in jedem Computersystem, die Rechenoperationen und verfügt in den meisten Fällen über eine Advanced RISC Machines (ARM)-Architektur. Die modernen Smartphone-Prozessoren werden zum Teil mit Acht-Kern-Prozessoren ausgestattet, die sehr viel Leistung und Energieeffizienz mit sich bringen. Die wesentlichen Unterschiede zwischen einem Computer und einem Smartphone sind, abgesehen von dem evidenten Größenunterschied, die zahlreich eingebauten Sensoren. Eine Klassifizierung und ein paar Beispiele sind in Tabelle 2.1 eingetragen.

Bewegungssensoren	Beschleunigungssensor	Gyroskop
Umgebungssensoren	Umgebungslichtsensor	Barometer
Positionssensor	Hall-Sensor	Magnetmeter
Authentifizierungssensor	Fingerabdrucksensor	Gesichtserkennung

Tabelle 2.1: Übersicht von Smartphone-Sensoren

Diese Sensoren erlauben dem Smartphone eine Vielzahl von weiteren Funktionen z. B. die Möglichkeit zu fotografieren oder zu filmen, Ton-aufnahmen oder -wiedergaben, Temperatur-, Druck-, Luftfeuchtigkeit- oder Herzpulsmessung. Des Weiteren kann die genaue Position durch das eingebaute Global Positioning System (dt. Globales Positionsbestimmungssystem) (GPS) bestimmt werden.

Ein weiteres Merkmal des Smartphones sind die relativ großen und hochauflösenden Bildschirme. Der Bildschirm bietet neben der Anzeigefunktionalität auch die Möglichkeit, mit dem Smartphone direkt zu interagieren ohne eine mechanische Tastatur zu verwenden. Dies wird durch die eingebauten Touch-Sensoren ermöglicht.

Auch im Bereich der Kommunikation sind die Smartphones mit der neuesten Technik ausgestattet, von den üblichen Kommunikationsprotokollen der normalen Mobiltelefonie, aufgelistet in der Tabelle 2.2 [18, S.64-66], bis hin zu weiteren verwendeten Verbindungsarten z. B. Wireless Local Area Network (dt. drahtloses lokales Netzwerk) (WLAN), Bluetooth oder Infrarot. Auch die beliebte Universal Serial Bus (USB)-Kabelverbindung aus der Computerwelt ist im Smartphone zu finden. Mittlerweile sind sogar die USB 3.0 Kabelverbindungen bei den neueren Smartphone-Modellen zu finden, die eine Datenübertragung mit Geschwindigkeiten von 5 Gbit/s erreichen können [14, S.3-4].

Kommunikationsprotokolle
GSM
HSCSD
UMTS
HPDSA
LTE

Tabelle 2.2: Übersicht Kommunikationsprotokolle

Für diese Arbeit wurde ein Smartphone des Herstellers Samsung verwendet und zwar das

Modell Galaxy S9 Edge. Das Gerät verfügt über einen sehr starken Prozessor mit 8 Kernen (Octa-Core), getaktet bei 2.7 GHz, einen Arbeitsspeicher von 4 GB Random-Access Memory (RAM), USB Typ-C, Near Field Communication (dt. Nahfeldkommunikation) (NFC) und einer Reihe weiterer Eingenschaften und Sensoren, die aber in dieser Arbeit nicht genannt werden.

2.1.2 Sensorik

Für die Datenaufzeichnung wird der Microelectromechanical systems (MEMS) Beschleunigungssensor verwendet. Dieser Sensor wird von der Firma STMicroelectronics hergestellt. Das verwendete Modell ist das LSM6DSL [20].

Das LSM6DSL ist ein Bauteil, das neben dem Beschleunigungssensor das Gyroscop beinhaltet. Für die beiden Sensoren erfolgt eine Signalaufzeichnung über die drei Ebenen X, Y, Z, wie in der Abbildung 2.1 zu sehen ist. Der Sensor wird in Bewegungsverfolgungssystemen angewendet. Außerdem findet er Anwendung in Funktionen wie intelligentes Strom sparen oder wird anstelle von GPS für die Navigation innerhalb von Gebäuden verwendet.

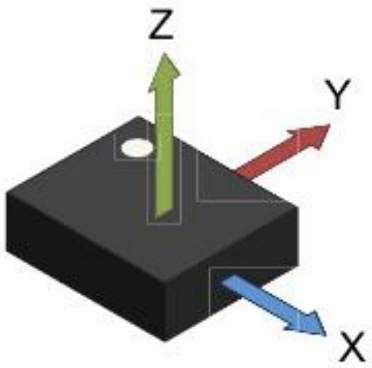

Abbildung 2.1: LSM6DSL 3-Achsen Übersicht [20, S.18]

Außer den vielen Einsatzmöglichkeiten bietet das LSM6DSL in dieser Arbeit einen weiteren Vorteil. Da der Sensor sehr intensiv betrieben wird, darf der Stromverbrauch nicht sehr hoch sein. Ein Verbrauch von 0.4 mA im Normalfall und 0.65 mA im Hochbetrieb ist für diese Anwendung am besten geeignet [20, S.1].

Ein weiterer interessanter Aspekt sind die Frequenzen, mit dem LSM6DSL die Daten liefert. Hier gibt es eine Auswahl mit festgelegten Frequenzwerten zwischen 1.6 Hz und 6.664 kHz, welche in der Theorie einer Zeit von bis zu 0.15 ms entsprechen [20, S.22].

Bei der Verwendung des Beschleunigungssensors wird in Ruhelage immer in der unteren Richtung (Z-Achse in der Abbildung 2.1) eine Beschleunigung von $a \approx 9.81 \frac{m}{s^2}$ stattfinden. Dieser Wert entspricht der Erdanziehungskraft auf diesem Gerät und ist überall auf der Erde gleich. Um diesen Offset loszuwerden, bietet der Sensor die Möglichkeit direkt mit einer linearen Beschleunigung zu arbeiten. Das entspricht in Ruhelage entlang der drei Achsen einer Beschleunigung von $a = 0 \frac{m}{s^2}$.

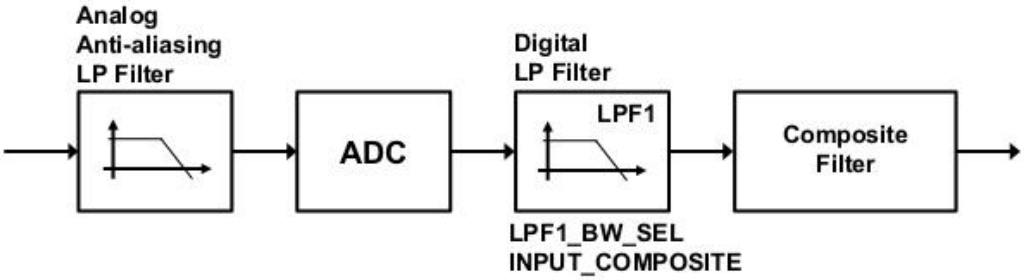

Abbildung 2.2: Übersicht LSM6DSL Beschleunigungssensorkette [20, S.33]

Nach der internen Wandlung der Analogwerte gemäß Abbildung 2.2 werden die Sensorwerte für jeweils eine der drei Achsen in dem entsprechenden Register als 16-Bit Zweierkomplementwert gespeichert. Auf diese Werte wird später in der Arbeit zugegriffen und mit deren Hilfe das Modell gebaut.

Die Abbildung 2.3 zeigt die Information über den verwendeten Sensor und die möglichen Einstellungswerte, die auf dem Samsung Galaxy S9 Smartphone verfügbar sind.

Abbildung 2.3: Sensorübersicht in Software auf dem Galaxy S9 Smartphone

Auf der linken Seite in Abbildung 2.3 sind Informationen über den Beschleunigungssensor z. B. Bezeichnung, Modell und Auflösung angegeben. Auf der rechten Seite sind die möglichen Betriebsfrequenzen und deren Grenzen angezeigt. Auch wenn der Sensor über andere Frequenzen betrieben werden kann, stellt die Software an dieser Stelle nur vier Einstellmöglichkeiten:

- SENSOR_DELAY_NORMAL ($\approx$ 5.5 Hz)
- SENSOR_DELAY_UI ($\approx$ 16.6 Hz)
- SENSOR_DELAY_GAME ($\approx$ 50 Hz)
- SENSOR_DELAY_FASTES ($\approx$ 500 Hz)

Die Sensorwerte beziehen sich auf Messungen mit dem Galaxy S9 Smartphone. Mehr über die ausgewählte Sensorfrequenz folgt im Kapitel 4.

2.2 Ganganalyse

Dieser Abschnitt dient dazu, den Leser mit Hintergründen der Ganganalyse vertraut zu machen. Die Ganganalyse ist für diese Arbeit ein sehr wichtiger und interessanter Aspekt, aus dessen Merkmalen das spätere Authentifizierungsverfahren gebildet wird.

Ziel der Ganganalyse ist, den Gang einer Person zu analysieren, zu untersuchen und zu beschreiben. Die Ganganalyse liefert unabhängige und quantifizierbare Messergebnisse.

Einsatzmöglichkeiten sind unter anderem in der Therapieüberwachung zu finden. Für die Sturzprävention können anhand der Ganganalyse mögliche Sturzrisiken bei Probanden gefunden, die Ursachen dazu analysiert und mögliche Maßnahmen zur Verminderung des Risikos entwickelt werden. Ein weiteres Anwendungsgebiet ist die Entwicklung und Verbesserung von Prothesen für die unteren Extremitäten. Letztendlich wird die Ganganalyse als Mittelpunkt des biometrischen Authentifizierungsverfahren eingesetzt, welches auch Teil dieser Arbeit darstellt.

Die natürlichste Art für den Menschen sich von A nach B fortzubewegen ist das Gehen. Dabei können unterschiedliche Hindernisse, Bodenverhältnisse und Neigungen bewältigt werden. Grund dafür ist die Evolution der unteren Extremitäten, die sich im Laufe der Zeit an diese Anforderungen angepasst haben. Das Gehen ist ausgesprochen reproduzierbar und verhält sich unter physiologischen Bedingungen bei allen gesunden Menschen nach demselben Grundmuster, das weitestgehend im Unterbewusstsein abläuft. Voraussetzung dafür ist jedoch eine uneingeschränkte Gelenkbeweglichkeit sowie ein dem Gang entsprechendes in Zeit und Intensität variables muskuläres Koordinationsmuster. Der menschliche Gang ist stark individuell und wird von einer Reihe Faktoren und Merkmalen geprägt. Eine wichtige Rolle in der Gangunterscheidung spielt unter anderem das Geschlecht, das Alter und das Gewicht [16, S.88-92]. Auch die Körpergröße, die Entwicklung der Muskulatur oder Vorschädigungen am Körper sind an dieser Stelle erwähnenswert.

Diese ganzen Unterschiede werden anhand eines definierten Bezugssystems, welches ein normales Gangbild darstellt, festgelegt. Da jede Person einzigartige Merkmale besitzt ist eine Unterscheidung nur anhand der Ganganalyse möglich.

Um ein besseres Bild zu bekommen wird als nächstes ein Gangzyklus analysiert und die Unterscheidungsmerkmale genauer erläutert. Dieser Begriff ist durch den Zeitraum definiert, der zwischen zwei aufeinanderfolgenden initialen Bodenkontakten desselben Fußes liegt [13, S.9].

Der erste Bodenkontakt wird als Anfang des Gangzykluses bezeichnet und somit als 0% Punkt des gesamtes Gangzykluses gekennzeichnet. Die Wiederholung des Bodenkontaktes mit demselben Fuß wird als Ende des Gangzykluses gekennzeichnet und gleichzeitig als 100% Punkt festgelegt.

Der Gangzyklus lässt sich in zwei Phasen unterteilen: Stand- und Schwungphase. Die Standphase wird als Phase bezeichnet, in der ein Fuß Bodenkontakt hat. Demzufolge ist die Phase bei dem das andere Bein in der Luft schwingt als Schwungphase definiert. Die Aufteilung der Phasen während des normalen Ganges erfolgt ca. 60% für die Standphase und ca. 40% für die Schwungphase.

Gehen ist im Unterschied zum Laufen durch eine Phase charakterisiert, in der beide Beine gleichzeitig am Boden stehen und das Körpergewicht auf jeweils ein Bein aufgeteilt wird. Beim Laufen berührt nur ein Bein den Boden, wodurch das ganze Körpergewicht somit nur auf diesem Standbein beruht.

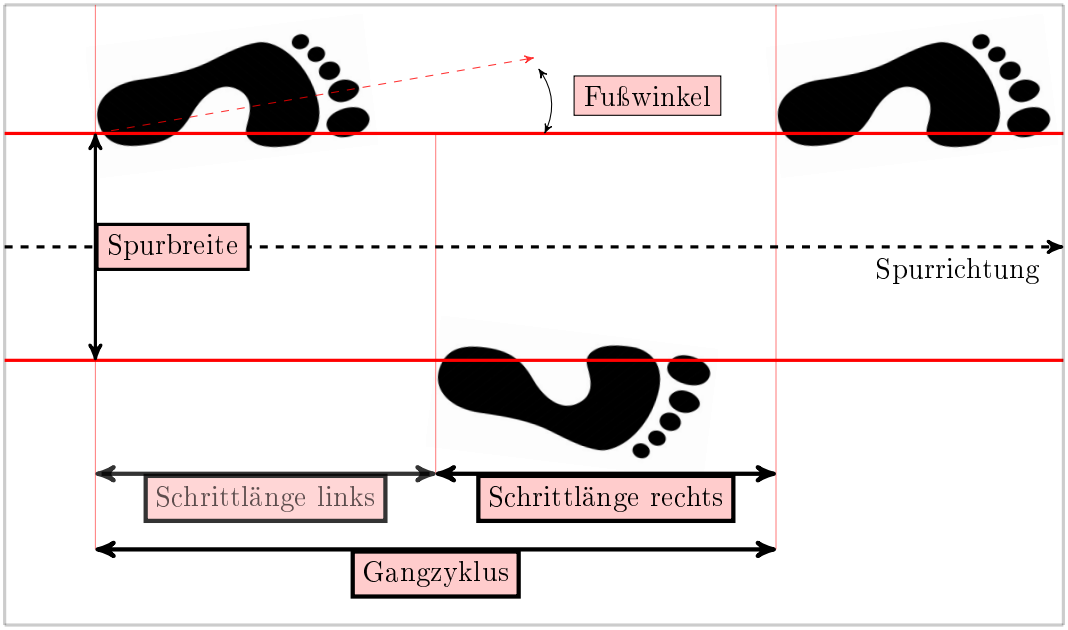

Abbildung 2.4: Übersicht des Gangzykluses

Bei der Untersuchung vom Gangzyklus sind folgende Merkmale aufgefallen und werden anhand der Abbildung 2.4 erläutert:

- Gangtempo

- Schrittlänge
- Spurbreite
- Fußwinkel

Das Gangtempo gibt Auskunft über die zurückgelegte Wegstrecke pro Zeit. Diese weist große Unterschiede auf. Das Gangtempo beträgt im Idealfall durchschnittlich 110 bis 120 Schritte pro Minute. Schnelleres Gehen kostet mehr Kraft während das langsamere Gehen die automatische Schrittauslösung unterdrückt.

Die Schrittlänge zeigt die Distanz zwischen den Kontaktstellen der beiden Füße an. Die Seitenzugehörigkeit einer Schrittlänge bezieht sich immer auf das Bein, welches nach Abschluss seiner Schwungphase initialen Bodenkontakt hat. Die Schrittlänge wird entlang der Fortbewegungslinie gemessen. Kürzere oder längere Schritte haben eine Auswirkung auf den Weggewinn. Damit die Schrittlänge vergleichbar beobachtet werden kann, muss man eine Schrittfrequenz von 110–120 Schritten pro Minute zugrunde legen, da sich die Schrittlänge mit unterschiedlicher Frequenz ebenfalls verändert.

Die Spurbreite wird anhand der Fersen beider Füße bestimmt. Die Entfernung wird senkrecht zur Fortbewegungslinie gemessen. Im Normalfall ist die Spurbreite etwas geringer als der Hüftgelenkabstand. Im Idealfall kann das Spielbein das Standbein ohne Berührung am Innenknöchel des Standbeins überholen.

Die virtuelle Fußachse ist eine gedachte Linie, die zusammen mit der Fortbewegungsrichtungsachse für das jeweilige Bein ein Winkel bildet, welches als Fußwinkel gekennzeichnet wird. Beim gesunden Menschen beträgt der Winkel ca. 11°.

Mit Hilfe dieser neuen Erkenntnisse lässt sich das Gangbild definieren als rhythmische und alternierende Wiederholung von mehreren Gangzyklen.

In Abbildung 2.5 wird nochmals das Gangbild mit den jeweiligen Abschnitten (Gangzyklen) anhand aufgezeichneter Sensordaten dargestellt. Diese Arbeit basiert auf einen vergleichbaren Versuch.

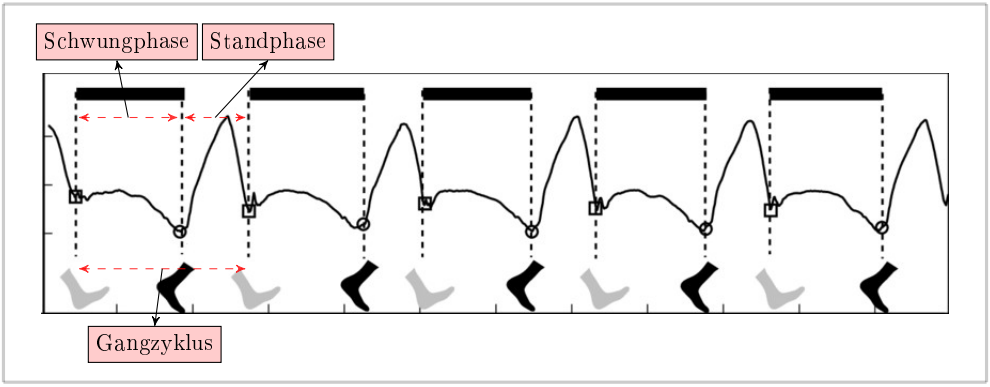

Abbildung 2.5: Untersuchung des Ganges mithilfe von Smartphone-Sensorik [3, S.2]

Die Stand- und Schwungphasen sind hier gut sichtbar. Es ist eine größere Änderung in der Schwungphase zu bemerken, in kürzester Zeit ist ein sehr hoher Impuls zu sehen, welcher aber dann auch sehr schnell wieder abfällt. In der Standphase ist die Änderung in der Amplitude nicht sehr groß, somit ist in dieser Phase der Verlauf der Kurve relativ flach. Die Aufteilung der Zeitspanne von ca. 60% für die Stand- und ca.40% für die Schwungphase lässt sich anhand der Abbildung 2.5 sehr gut beobachten.

2.3 Authentifizierung

Der folgende Abschnitt behandelt die notwendigen Grundlagen für das weitere Verständnis der Arbeit und klärt in diesem Zusammenhang wichtige Begriffe zum Thema Authentifizierung.

Zunächst werden die wichtigsten Begriffe der Authentifizierung definiert und verschiedene Verfahren klassifiziert. Hieran schließt sich unter anderem eine Definition, eine Klassifikation sowie eine Beschreibung des Aufbaus und der typischen Funktionsweise biometrischer Authentifizierungssysteme.

2.3.1 Klassifikation der Authentifizierung

Die Authentizität bezeichnet die Echtheit und Glaubwürdigkeit des Objekts beziehungsweise (bzw.) Subjekts, die anhand einer eindeutigen Identität und charakteristischen

Eigenschaften überprüfbar ist. Die Authentifizierung stellt die Maßnahmen zur Überprüfung dieser Authentizität dar und erfolgt, indem eine behauptete Identität anhand seiner charakterisierenden Eigenschaften nachgewiesen werden kann. Die Identifikation basiert auf der Vergabe von einer eindeutigen Benutzerkennung oder eines Benutzernamens. Ein Beispiel von charakterisierenden Eigenschaften zum Nachweis der Identität sind Passwörter, deren Kenntnisse nur dem Nutzer bekannt sind, oder biometrische Merkmale, die eindeutig auf einen bestimmten User nachgewiesen sind. Solche Identitätsnachweise werden häufig unter dem Begriff Credentials verwendet[10, S.8]. Die Aufgabe der Sicherheitsgrundfunktion der Authentifizierung besteht darin, mit geeigneten Maßnahmen die Richtigkeit einer behaupteten Identität einschätzen zu können. Die Klassifizierung des Authentifizierungsverfahrens lässt sich grundsätzlich in drei Gruppen einteilen:

- Wissen
- Besitz
- Biometrie

Die Authentifizierung auf Basis von spezifischem Wissen ist am häufigsten in der Praxis verbreitet. Dieses Verfahren basiert auf dem sogenannten Challenge Response-Verfahren. Hier erfolgt die Authentifizierung durch die Kenntnis von einem Geheimnis. Das meist bekannte Authentifizierungsverfahren anhand von Wissen ist das Passwortverfahren. Hier erfolgt die Identitätsprüfung, indem der User die Kenntnis eines mit dem System vereinbarten Geheimnisses nachweisen kann. Um die Sicherheit zu garantieren, muss das vereinbarte Kennwort von dem Benutzer geheim gehalten werden. Auch das System muss bei der Verwaltung von Passwörtern die Garantie nachweisen, dass niemand außer dem autorisierten Benutzer an diese Passwörter gelangen kann. Hier wird i.d.R. auf eine kryptografische Hashfunktion gesetzt. Dazu wird aus dem Passwort ein kryptografischer Hashwert berechnet und statt des eigentlichen Passwortes gespeichert. Dies hat den Vorteil, dass eine direkte Wiederherstellung des ursprünglichen Passwortes aus der Kenntnis des Hashwertes ohne großen Aufwand nicht möglich ist.

Der Faktor Besitz bezeichnet die Methode, dass eine Person nur dann in der Lage ist sich authentifizieren zu können, wenn sie im Besitz eines bestimmten Objekts ist. Das trivialste Beispiel hierfür stellt der Schlüssel dar. Weitere Beispiele sind Firmenausweise oder Kreditkarten. Ein großer Nachteil bei diesem Verfahren ist, dass bei der Authentifizierung das Objekt immer dabei sein muss. Ein weiterer Nachteil besteht, dass sich bei Verlust oder Diebstahl nicht-befugte Personen fälschlicherweise authentifizieren können.

Um solch ein Szenario zu vermeiden wird oft eine weitere Schicht von Sicherheit zugefügt, zum Beispiel ein zusätzlicher PIN (Multi-Faktor-Authentifizierung).

Schließlich setzen biometrische Authentifizierungsverfahren auf personenbezogene Merkmale zur Verifikation oder Identifikation. Unter einem solchen Merkmal versteht man physiologische oder verfahrenstypische Eigenschaften einer Person, die diese eindeutig charakterisieren kann. Ein Vorteil für das biometrische Authentifizierungsverfahren ist, dass im Gegensatz zum Faktor Wissen und Besitz, die Biometrie nicht vergessen, verloren oder gestohlen werden kann. Die biometrischen Techniken zur Authentifizierung können selber in zwei Kategorien eingeteilt werden:

- Statisch
- Dynamisch

Der Begriff Statisch beruht auf den physiologischen Merkmalen einer Person z. B. Fingerabdrücke, das Gesicht oder die Iris, während sich Dynamisch auf die verfahrenstypischen Eigenschaften einer Person bezieht. Beispiele für Dynamik sind unter anderem die Stimme oder die Untersuchung von Tippverhaltensmuster einer Person.

Die Systemsicherheit ist im weitesten Sinne an die Anforderungen der biometrischen Authentifizierungsmethode gekoppelt. Allgemein existiert eine Vielzahl von verschiedenen Anforderungen, die ein Authentifizierungssystem mindestens erfüllen soll. In der Literatur werden wie folgt sieben unterschiedliche Anforderungen für die biometrische Authentifizierung definiert[10, S.496]:

- Universalität
- Eindeutigkeit
- Beständigkeit
- Quantitative Erfassbarkeit
- Performanz
- Akzeptanz
- Fälschungssicherheit

Universalität: Das biometrische Authentifizierungsmerkmal sollte generell bei allen Personen existieren. Wenn nur eine geringe Anzahl an Personen das Verfahren nutzen kann, ist das Authentifizierungsverfahren nicht einsetzbar oder nur begrenzt zu verwenden.

Eindeutigkeit: Die Charakteristik sollte über ausreichend individuelle Eigenschaften in der Bevölkerung verfügen. Dabei muss eine Person duech seine Charakteristiken eindeutig erkennbar sein. Es darf maximal eine Person dieses biometrische Merkmal besitzen.

Beständigkeit: Das Merkmal sollte sich im Laufe der Zeit in einem möglichst geringen Umfang verändern. Wenn sich ein Merkmal über die Zeit stark verändert, ist es als biometrische Eigenschaft für das Authentifizierungsverfahren ungeeignet.

Quantitative Erfassbarkeit: Das biometrische Verhalten sollte mit technischen Möglichkeiten quantitativ messbar sein.

Performanz: Die Genauigkeit und Performanz bei der Erfassung von Merkmalen soll einen bestimmten vorgegebenen Standard nicht unterschreiten.

Akzeptanz: Die Messung des Merkmals soll vom Anwender akzeptiert und leicht zu bedienen sein.

Fälschungssicherheit: Die Fälschung von Merkmalen soll nicht möglich bzw. nicht ohne erhebliche Schwierigkeiten durchzuführen sein.

Zusätzlich zu diesen sieben Begriffen hat es sich als sinnvoll herausgestellt auch weitere Anforderungen zu stellen. Da die Sicherheit während der gesamten Nutzung gewährleistet sein muss und nicht nur zum Zeitpunkt der initialen Authentifizierung, wenn beispielsweise das Entsperren eines Smartphones durchgeführt wurde, wird an dieser Stelle der Begriff von **Kontinuität** eingeführt.

Geräteunabhängigkeit ist eine weitere Anforderung welche eingehalten werden muss. Durch die Vielfalt an Geräten mit jeweils verschiedenen Sensoren ist es wichtig, dass eine Authentifizierung generell und unabhängig davon, welches Gerät verwendet wird, gleich gut funktioniert.

Als Letztes wird in dieser Arbeit der Begriff **Unabhängigkeit** präsentiert. Es ist relevant, dass für die Authentifizierung keine Zusatzgeräte benötigt werden. Diese würden die Anforderung der Akzeptanz negativ beeinflussen.

2.3.2 Aufbau und Verfahrensablauf biometrischer Systeme

Biometrische Systeme arbeiten alle nach dem gleichen Schema. An der ersten Stelle stehen technische Einrichtungen zur Erfassung und Digitalisierung des individuellen biometrischen Merkmals. Diese Aufgaben werden i.d.R. von der Sensorik übernommen. Die Art des Merkmals ist ausschlaggebend dafür welcher Sensor verwendet wird. Anschließend abstrahieren mathematische und statistische Methoden die erfassten Sensordaten und speichern sie als personenbezogene Referenzdatensätze in einem Template. Zum Schluss werden schließlich die Vergleichsalgorithmen erstellt. Der Aufbau eines biometrischen Systems ist in der Abbildung 2.6 [9, S.16] dargestellt.

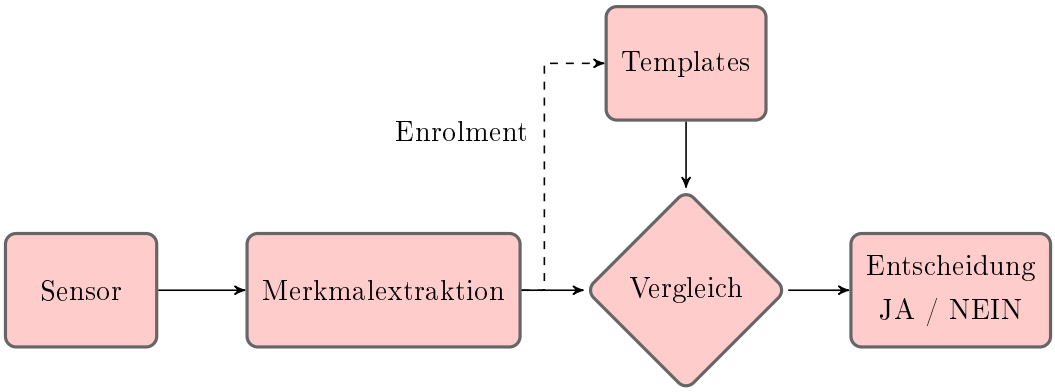

Abbildung 2.6: Aufbau eines biometrischen Systems

Es existieren im kompletten Verfahrensablauf zwei unterschiedliche Phasen. In der ersten Phase werden Referenzmerkmale von einer Person in der Datenbank gespeichert. Diese Phase wird als **Enrolment** bezeichnet. In der zweiten Phase wird die Person authentifiziert, damit sie den Zugriff zu einem System erhält. Die zweite Phase ist unter **Verifikation** bekannt. Beide Phasen bestehen aus mehreren Schritten. Abbildung 2.7 zeigt, dass sowohl bei dem Enrolment als auch der eigentlichen Authentifizierung vier Schritte existieren, die während beider Phasen ausgeführt werden. Dazu zählen die Datenaufnahme, Vorverarbeitung, die Extraktion der Merkmale und die Erstellung eines Templates. Ab diesem Punkt wird bei Enrolment das Template als Referenzdatei gespeichert. Das Template stellt das Referenzprofil des Nutzers dar, welches sich meist aus einer Kombination mehrerer Messungen zusammensetzt und anschließend mit der Identität des Benutzers verknüpft wird.

In der zweiten Phase (Verifikation) erfolgt nach der Erstellung des Templates ein Ver-

gleich mit dem gespeicherten Template. Der Grad der Übereinstimmung wird durch einen Wert ausgedrückt. Dieser Wert wird in Abbildung 2.7 durch den Matching-Score-Block repräsentiert. Da es praktisch fast unmöglich ist auch bei derselben Person eine 100-prozentige Übereinstimmung zu erzielen, werden Schwellen festgelegt.

Das Matching-Score wird zunächst mit der zuvor bestimmten Akzeptanzschwelle verglichen. Je nachdem, ob der Wert der Akzeptanzschwelle erreicht wurde, trifft das System dann eine Entscheidung ob eine Übereinstimmung, keine Übereinstimmung oder kein Ergebnis vorliegt.

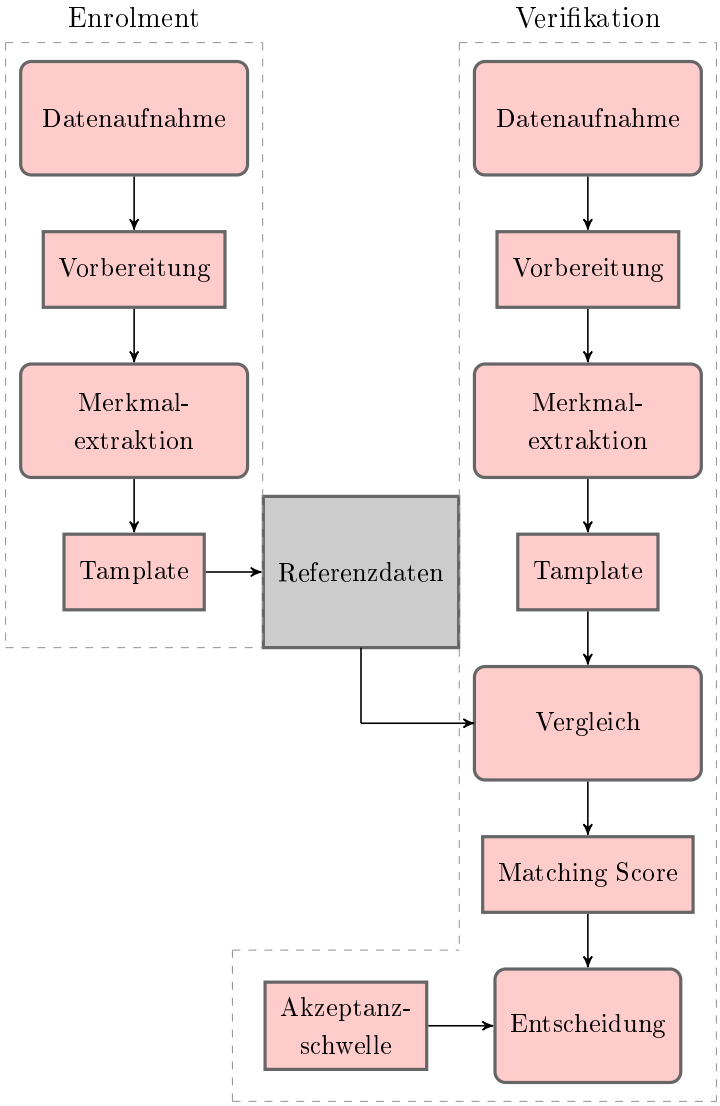

Abbildung 2.7: Komponenten und Ablauf eines biometrischen Verfahrens

Datenaufnahme und Vorbereitung

Die Datenaufnahme erfolgt mit Hilfe von Sensoren. Es werden über eine bestimmte Zeitperiode oder sogar durchgehend sogenannte Samples aufgenommen. Der Aufnahmeprozess kann auch erst nach einem Trigger stattfinden. Ein Sample ist eine analoge oder digitale Repräsentation einer biometrischen Eigenschaft. Die aufgenommenen Samples werden dann in einer Datei oder Datenbank gespeichert.

Da die Daten nicht immer in der gleichen Qualität von den Sensoren aufgenommen werden und meistens im biometrischen Sample mehr Informationen enthalten sind als für die Merkmalextraktion benötigt wird, werden die Daten an den Verarbeitungsprozess weitergereicht. Hier werden die Daten bereinigt. Dazu werden passende Filter benutzt, um unnötige Information oder Rauschen zu entfernen. Die überflüssige Information stört an dieser Stelle nicht, wird sich aber später in der Bearbeitungszeit bemerkbar machen. Da die Verarbeitungszeit einen direkten Einfluss auf die Akzeptanz hat, soll die Verarbeitungszeit so klein wie möglich gehalten werden.

Da es nicht sinnvoll ist mit einem langen Datensatz zu arbeiten, werden die Samples in gleichlange Segmente unterteilt. Dies ermöglicht eine parallele Verarbeitung und erspart dadurch Zeit. Wenn die Qualität der Daten trotz Verarbeitung zu gering ist, wird der Datensatz verworfen und der Aufnahmeprozess wiederholt.

Merkmalextraktion und Template erstellen

Die Extraktion von Merkmalen stellt die Generierung von Eigenschaften aus den Daten und einem zentralen Punkt während des Authentifizierungsprozesses dar, da die Fehlerraten stark von der geeigneten Auswahl der Merkmale abhängen. Daher wird im Abschnitt 3.4 genauer auf diese Merkmale und der Templateerstellung eingegangen.

Klassifikation und Entscheidung

Bei der Klassifikation für die biometrische Authentifizierung handelt es sich um eine Funktion. Mit Hilfe dieser Funktion wird ein Objekt mit einer Klasse aus einer Menge verknüpft. Die Menge wird mit Hilfe von Modellen einzelner Personen dargestellt, die während des Enrolmentprozesses trainiert wurden. Anhand dieser Klassifizierung wird entschieden, ob die Person den Zugriff erhält oder abgewiesen wird. Zur Auswahl gibt

es eine Reihe von Klassifikatoren, zum Beispiel Support Vector Machines (dt. Stützvektormaschine) (SVM), neuronale Netze oder Entscheidungsbäume. Da die Klassifikatoren von der Merkmalauswahl abhängig sind, werden im Abschnitt 3.5 die ausgewählten Klassifikatoren anhand echter Daten genauer untersucht.

2.3.3 Genauigkeit

Die Genauigkeit Accuracy (dt. Genauigkeit) (ACC) besagt, wie gut ein Klassifikator zwischen mehreren Klassen unterscheiden kann. Zusätzlich besagt die Genauigkeit, wie gut ein Klassifikator neue Testwerte zu einer Klasse zuordnen kann. Die Genauigkeit wird folgendermaßen errechnet:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} \tag{2.1}$$

TP - Anzahl aller Anmeldungen von richtig autorisierten Personen
TN - Anzahl aller Anmeldungen von richtigen nicht autorisierten Personen
P - Anzahl aller Anmeldungen von autorisierten Personen
N - Anzahl aller Anmeldungen von nicht autorisierten Personen

Die Genauigkeit kann Werte zwischen null und eins bzw. 0% und 100% annehmen. Das beste Ergebnis wird mit einer Genauigkeit von 100% erreicht, während das schlechteste Ergebnis mit einem Wert von 0% gekennzeichnet wird.

2.3.4 Fehlerraten

Ein großer Nachteil des biometrischen Authentifizierungsverfahrens ist die Tatsache, dass sie nicht immer exakt sind, zum Beispiel bei der Authentifizierung anhand einer PIN oder eines Passwortes. Hier sind einfache Tests auf Gleichheit ausreichend, um eine 100-prozentige Genauigkeit bei einer Übereinstimmung zu erzielen. Bei biometrischen Merkmalen hingegen ist die Sache etwas komplizierter. Hier werden die Eigenschaften einer Person untersucht und diese können sich mit der Zeit oder unter bestimmten Umständen z. B. Wetter, Lichtverhältnis oder Verschmutzung verändern. Das führt dazu, dass sich die aktuellen Werte häufig von den Referenzwerten leicht unterscheiden.

Eine Lösung hierfür ist, eine Toleranzschwelle zu definieren und mittels Korrelationstest die Abweichungen von den Referenzwerten zu bestimmen. Die Abweichung kann als Fehler interpretiert werden und die Abweichungen vom Referenzwert als Fehlerraten. An dieser Stelle werden die Begriffe von False Acceptance Rate (dt. Falschakzeptanzrate) (FAR), False Rejection Rate (dt. Falschrückweisungsrate) (FRR) und Equal Error Rate (dt. Gleichfehlerrate) (EER) vorgestellt. In Abbildung 2.8 [10, S.500-501] wird der Zusammenhang zwischen diesen Fehlerraten veranschaulicht.

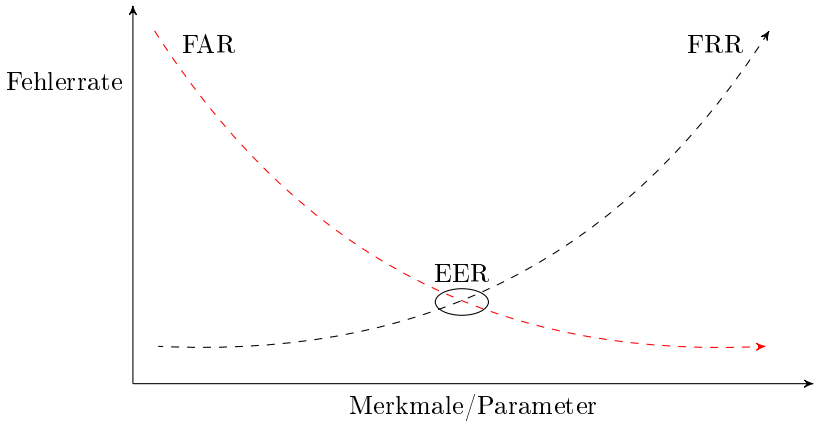

Abbildung 2.8: Zusammenhang zwischen FAR, FRR und EER

Grundsätzlich wird zwischen zwei Fehlern unterschieden. Der erste Typ von Fehler: ein berechtigter Benutzer wird abgewiesen. Wenn dieser Fehler auftritt sind meistens die Kontrollmaßnahmen zu streng. Eine Anpassung der Schwellenwerte wird hier benötigt.

Der zweite Typ von Fehler: ein unberechtigter Benutzer wird authentifiziert und akzeptiert. Das Auftreten des Fehlers soll möglichst komplett vermieden werden, da hier unbefugte Personen Zugang zu Systemen bekommen. Die Auftrennung der ersten Fehler mit einer sehr hohen Häufigkeit kann dazu führen, dass sich die Akzeptanz verschlechtern kann. Die Wahl des Schwellenwertes hat gegenläufige Konsequenzen für FAR und FRR, daher muss ein Gleichgewicht gefunden werden.

False Acceptance Rate (FAR)

Die FAR gibt die Wahrscheinlichkeit an, wie viele Angreifer sich an dem System anmelden können. Deshalb ist diese Fehlerrate ein Maß für die Sicherheit des Systems. Diese liegt zwischen 0% und 100%.

$$\text{FAR} = \frac{\text{Anzahl an falsch akzeptierten Personen}}{\text{Anzahl aller Anmeldeversuche unautorisierter Personen}} \tag{2.2}$$

Je mehr Angreifer sich authentifizieren können, desto größer wird die FAR. Ein typischer Wert ist:

$$\text{FAR} = 1 \cdot 10^{-5}$$

Das bedeutet, dass im statistischen Mittel einer von 100.000 unberechtigten Zugangsversuchen zum Erfolg führen wird.

False Rejection Rate (FRR)

Im Gegensatz zur FAR gibt FRR die Wahrscheinlichkeit an, mit der eine berechtigte Person abgewiesen wird. Diese Fehlerrate repräsentiert Informationen über die Benutzerfreundlichkeit des Systems. Auch der Wertebereich von FRR liegt zwischen 0% und 100%.

$$\text{FRR} = \frac{\text{Anzahl an falsch zurückgewiesenen Personen}}{\text{Anzahl aller Anmeldungen von autorisierten Personen}} \tag{2.3}$$

Ein typischer Wert ist:

$$\text{FRR} = 1 \cdot 10^{-3}$$

Dieser Wert besagt, dass von 1000 berechtigten Nutzern im statistischen Mittel nur einer fälschlicherweise abgewiesen wird.

Equal Error Rate (EER)

Die EER ist der Schnittpunkt von den FAR und FRR Kurven wie in der Abbildung 2.8 zu sehen ist. Es ist die Zahl, bei der die Anzahl der fälschlich abgewiesenen Benutzer und die Anzahl der fälschlich akzeptierten Benutzer gleich groß ist. Die EER kann mit folgenden Formel errechnet werden:

$$\mathrm{EER} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{P} + \mathrm{N}} \tag{2.4}$$

FP - Anzahl an falsch zurückgewiesenen Personen
FN - Anzahl an falsch akzeptierten Personen
P - Anzahl aller Anmeldungen von autorisierten Personen
N - Anzahl aller Anmeldungen von nicht autorisierten Personen

Der Wert von EER liegt zwischen null und eins. Der beste EER-Wert ist null und schlechteste ist eins.

Die Gleichfehlerrate wird heute als aussagekräftigstes Maß für die Güte eines biometrischen Systems angesehen. Sie berücksichtigt allerdings nur das Verhalten des Systems in einem kritischen Punkt und vernachlässigt den Verlauf von FAR und FRR bei anderen Fehlerraten.

Die drei Fehlerraten FAR, FRR und EER dienen als Leistungsmaß zur Bewertung der Güte eines biometrischen Erkennungssystems.

2.4 Software

In diesem Abschnitt wird die nötige Software für die Realisierung dieser Arbeit vorgestellt.

2.4.1 MATLAB

MATLAB ist ein Software-Tool zur Lösung, Berechnung und Simulation von komplexen numerischen, mathematischen und technischen Problemen. Die Software bietet auch diverse Möglichkeiten zur grafischen Visualisierung und Darstellung von Ergebnissen.

MATLAB ist eine Abkürzung für MATRIX LABORATORY und wurde in den 1970er von Clive Moeler und Jack Little entwickelt. Die Software wird von deren in der 1980 gegründeten Firma The MathWorks.Inc vertrieben [2].

MATLAB ist eine englischsprachige Software und benutzt englische Begriffe und Befehle. Der Hersteller bietet zusätzlich eine sehr gute Dokumentationsplattform mit Begriffen, Formeln, Algorithmenbeispielen, Schulungen und Webseminaren. Diese haben sich für den Einstieg und auch für erfahrene User als sehr hilfreich herausgestellt. Hier findet alles in englischer Sprache statt. Die Basis-Programmiersprachen hinter MATLAB sind Fortran and C. Der Basis-Datentyp in MATLAB ist die Matrix. Matrizen- und Vektorrechnungen sind deshalb einer der Stärken dieser Software. Die grafische Darstellung ist an dieser Stelle auch erwähnenswert. Mit einer Reihe von Darstellungsmöglichkeiten, Achsenskalierungen und Beschriftungen sowohl der Zeit- als auch Frequenzanalyse bietet MATLAB eine Lösung in der Industrie und an vielen Hochschulen für vielfältige Aufgaben z. B. Bild- und Signalverarbeitung, Finanzmodellierung und Analyse.

Zusätzlich zum Standardmodell bietet MATLAB mit den sogenannten TOOLBOXEN eine ganze Reihe von Erweiterungen, die ständig weiterentwickelt werden. Die Software erhält i.d.R. jährlich zwei große Updates. Ein Update erfolgt im Frühjahr, mit (a) gekennzeichnet und ein Update erfolgt im Herbst, mit (b) gekennzeichnet. Für diese Arbeit spielt MATLAB eine wichtige Rolle. Das Kapitel 3 basiert fast komplett auf dieser Programmiersprache. Die Datendarstellung und Visualisierung von Ergebnissen im Kapitel 3 sind alle in MATLAB Version 2018b realisiert. Hierfür wurde zusätzlich die Erweiterung Statistics and Maschine Learning Toolbox benutzt.

2.4.2 JAVA

Die Programmiersprache JAVA ist heute eine der meistverwendeten Programmiersprachen.

JAVA ist eine eingetragene Marke des Unternehmens Oracle und eine weitverbreitete Entwicklungsplattform.

Diese Programmiersprache wird als objektorientierte Sprache beschrieben. Hier wird versucht, unsere reale Welt möglichst einfach durch interagierende Objekte in eine Programmiersprache abzubilden. Diese Eigenschaft und die umfangreiche Kollektion von JAVA-Klassenbibliotheken, die den Anwendern das Programmieren wesentlich vereinfachen, machen aus JAVA die perfekte Programmiersprache für Einsteiger.

JAVA wurde von Beginn an mit dem Ziel entworfen, verteilte Anwendungen zu unterstützen. Diese erlaubt JAVA-Programme über die Transmission Control Protocol/Internet Protocol (TCP/IP) Socket, World Wide Web (dt. Weltweites Netz) (WWW)-Seiten zu lesen, Daten im Internet zu öffnen und zu bearbeiten. Diese ganzen Eigenschaften führen dazu, dass in JAVA vollständige Client-Server-Applikationen entwickelt werden können.

Ein großer Vorteil der JAVA Programmiersprache ist die Architekturunabhängigkeit. Programme, die mit JAVA entwickelt wurden, lassen sich auf jede Rechner-Architektur, bei der ein JAVA-VM vorhanden ist, ohne weitere Schwierigkeiten ausführen.

Einen weiteren Vorteil der JAVA Programmiersprache ist die Nebenläufigkeit. Diese Eigenschaft erlaubt die parallele Bearbeitung von mehreren Programmteilen, die somit die Laufzeit reduzieren können. Diese parallelen Pfade sind als Threads bezeichnet. Die Fähigkeit von Nebenläufigkeit wird als Multithreading angegeben [19, S.9-21].

Ein JAVA Programm funktioniert wie folgt:

Zuerst wird das Programm als Quelltext in einem Editor erstellt und dann als Quelltextdatei mit der Endung `.java` gespeichert.

Im nächsten Schritt erzeugt der Compiler sogenannte JAVA-Byte-Codes und wird in Dateien mit der Endung `.class` gespeichert. JAVA-Byte-Code-Dateien können nicht direkt vom Rechnersystem ausgeführt werden. Ab hier wird die JAVA-VM benötigt.

Die JAVA-VM ist die Schnittstelle zwischen dem JAVA-Byte-Code und der spezifischen Hard- und Software des ausführenden Systems und sorgt für die Ausführung von JAVA-Programmen. Die JAVA-VM führt folgende Schritte durch:

Zunächst wird der JAVA-Byte-Code geladen, dann wird nach weiteren im Programm benötigten Klassen gesucht. Wenn nötig werden die fehlenden Klassen aus Klassenbibliotheken geladen. Dieser Block wird als Byte-Code-Loader bezeichnet. Die nächste Aufgabe

wird von dem Byte-Code-Verifizierer übernommen. Wie der Name schon sagt, wird hier auf Verstöße gegen die JAVA-Sprachregeln verifiziert.

Anschließend wird der Byte-Code interpretiert und als maschinenspezifischer Code auf dem System ausgeführt. Eine Übersicht der JAVA-VM ist in Abbildung 2.9 dargestellt.

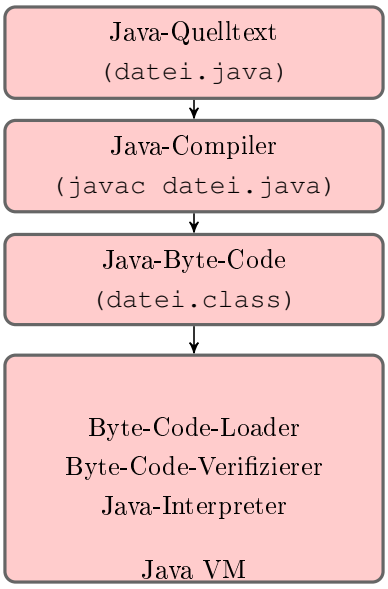

Abbildung 2.9: Java-VM [19, S.19]

Mit JAVA lässt sich eine große Anzahl von Softwareprojekten entwickeln wie Anwendungsprogramme, Gerätesteuerung, Computerspiele oder für die Entwicklung von ANDROID System-basierten Applikationen.

JAVA ist die Grundprogrammiersprache für die Implementierung des verwendeten Programms in dieser Arbeit.

2.4.3 ANDROID STUDIO

ANDROID ist eine freie Integrated Development Environment (dt. Integrierte Entwicklungsumgebung) (IDE) und wurde am 16. Mai 2013 auf der Entwicklerkonferenz von Google vorgestellt. Es ist die offizielle Entwicklerumgebung für die Erstellung und Programmierung von ANDROID-Applikationen. ANDROID STUDIO unterstützt alle Programmiersprachen die in IntelliJ vorhanden sind z. B. JAVA, Kotlin, XML/XSL, Python und

andere. Für die Implementierung aus Kapitel 4 wurde Java als Programmiersprache ausgewählt. Android Studio bietet eine Entwicklungsplattform für alle Android-Geräte. Ein großer Vorteil der Software ist die Möglichkeit, Programme auf einem virtuellen Gerät zu testen und Änderungen im Programmcode schnell zu aktualisieren ohne dafür ein Android Package Kit (dt. Android-Programmpakete) (APK) bauen zu müssen.

Projektstruktur in Android Studio

Jedes Projekt besitzt ein oder mehrere Module, welche Codedateien und Ressourcendateien beinhalten. Folgende Module sind vorhanden:

- Android App-Module
- Bibliotheks-Module
- Google App Engine-Module

Die Abbildung 2.10 zeigt den Aufbau eines Projektes in Android Studio. Auch hier gibt es mehrere Ansichtsmöglichkeiten. Als Standard liefert das Programm die sogenannte Android-Ansicht. Diese Ansicht ist in der Abbildung 2.10 dargestellt.

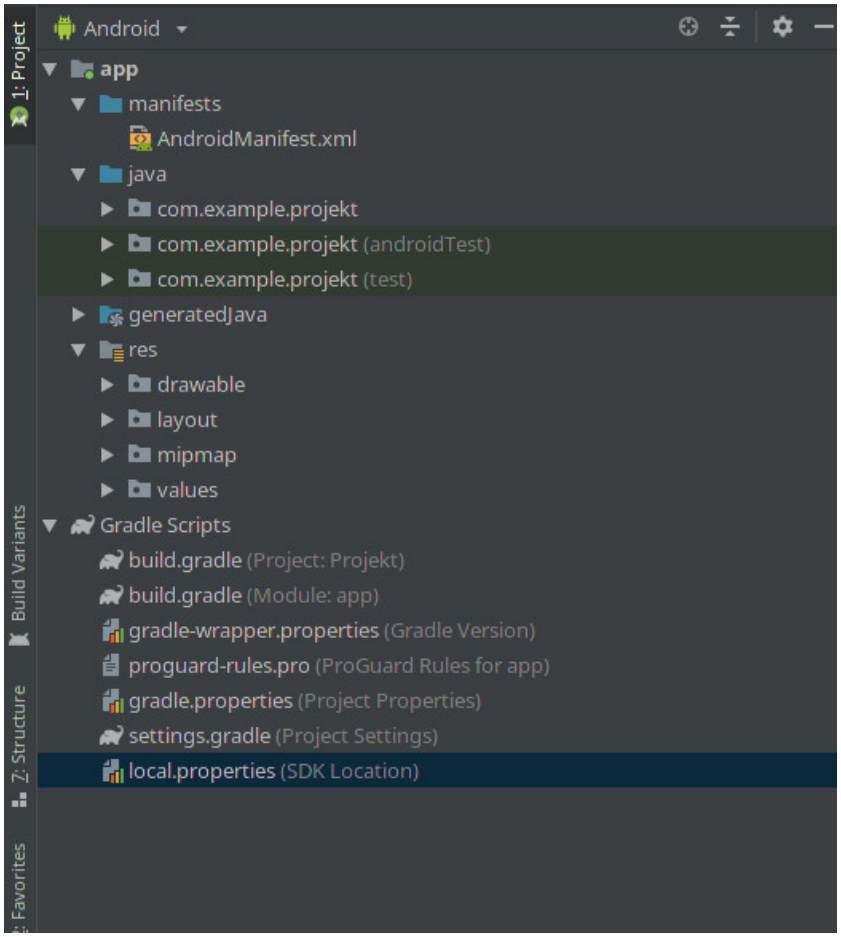

Abbildung 2.10: Projektstruktur in Android Studio

Das Android Applikationsmodul ist folgendermaßen aufgebaut:

- **manifest:** beinhaltet die `AndroidManifest.xml` Datei.
- **java:** beinhaltet den Java Quellcode und die Junit Test-Dateien.
- **res:** speichert alle externen Dateien zum Beispiel XML Layouts, UI Strings und Bilder.

In **Gradle Scripts** befinden sich Daten und Einstellungen für den APK Aufbau [1].

2.4.4 Weka Library

Waikato Environment for Knowledge Analysis (WEKA) ist ein Software-Tool, das verschiedene Techniken aus den Bereichen maschinelles Lernen und Data-Mining bereitstellt. Die Software wurde von der University of Waikato in Neuseeland entwickelt. Der Hersteller stellt zusätzlich zur Software eine Reihe von Beispielen, Dokumentationen und Büchern zur Verfügung. Die Sachen sind frei zugänglich und können auf der Internetseite des Herstellers gefunden werden (`https://www.cs.waikato.ac.nz/ml/weka/index.html`).

Die WEKA Software verfügt über eine Java-Bibliothek, die im weiteren Verlauf für die Implementierung verwendet wird.

Ein Vorteil von WEKA ist die Möglichkeit, die Software über eine Command Line Interface (dt. Kommandozeileninterpreter) (CLI) und ein Graphical User Interface (dt. Grafische Benutzeroberfläche) (GUI) zu betreiben. Besonders hilfreich hat sich die GUI Oberfläche für diese Arbeit herausgestellt. Eine Übersicht der GUI Oberfläche ist in Abbildung 2.11 dargestellt.

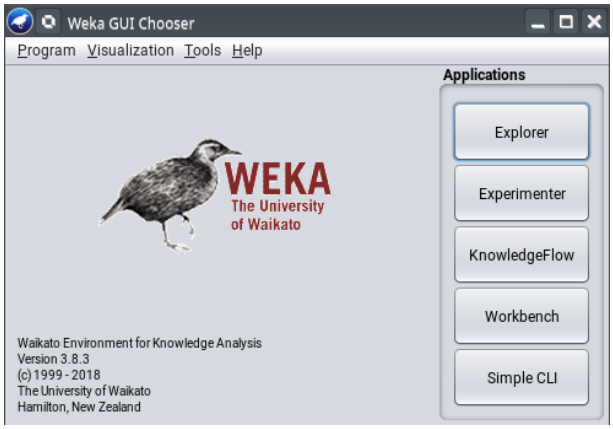

Abbildung 2.11: WEKA GUI-Oberfläche

Hier ist es möglich direkt mit Datensätzen zu arbeiten und mit nur wenigem Aufwand viele Informationen über die aufgezeichneten Sensordaten zu erfahren. Es ist auch möglich unterschiedliche Filter direkt anzuwenden und die Ergebnisse gleich in einem Graphen anzeigen zu lassen.

Die Software verwendet für die Eingabe von Datensätzen ein eigenes Dateiformat `.arff`. Attribute-Relation File Format (ARFF) ist eine American Standard Code for Information Interchange (ASCII) Textdatei, welche eine Liste von Beobachtungen und deren gemeinsamen Eigenschaften beschreibt [24, S.53-60]. Zusätzlich bietet Weka die Möglichkeit das weitverbreitete Comma-Separated Values (`.csv`) in das weniger bekannte `.arff` Format zu konvertieren. Die Konvertierung funktioniert in beide Richtungen. Ein weiterer und sehr wichtiger Vorteil der Software ist die große Auswahlmöglichkeit an Klassifikatoren wie Bayes-Klassifikatoren, künstliche neuronale Netze, Support-Vector-Maschinen, Entscheidungsbäume und viele andere.

3 Modellierung in Matlab

3.1 Vorgehensweise

In diesem Kapitel werden mit Hilfe der Software MATLAB die wichtigsten Prozesse bei der Erstellung eines biometrischen Verfahrensmodells untersucht und erläutert. Anhand extrahierten biometrischen Datensätzen wird Schritt für Schritt der Ablauf eines biometrischen Verfahrens aus Abschnitt 2.3.2 und Abbildung 2.7 gezeigt. Die Raw-Sensordaten werden verarbeitet, passende Eigenschaften (Merkmale) werden für die Sensorwerte ausgewählt und Templates (Merkmal-Tabellen) werden erstellt. Diese Schritte werden mit Hilfe von Texten, Formeln und MATLAB-Plots veranschaulicht.

Danach werden diverse Klassifikatoren untersucht und anhand der Verarbeitungsgeschwindigkeit und Genauigkeit ausgewertet.

Zum Abschluss des Kapitels erfolgt eine Auswertung der Ergebnisse anhand des erstellten MATLAB-Modells. Die Parameter von den besten Modellen, werden für die Implementierung übernommen.

Eine Übersicht der Matlab-Modellierung ist in der Abbildung 3.1 dargestellt.

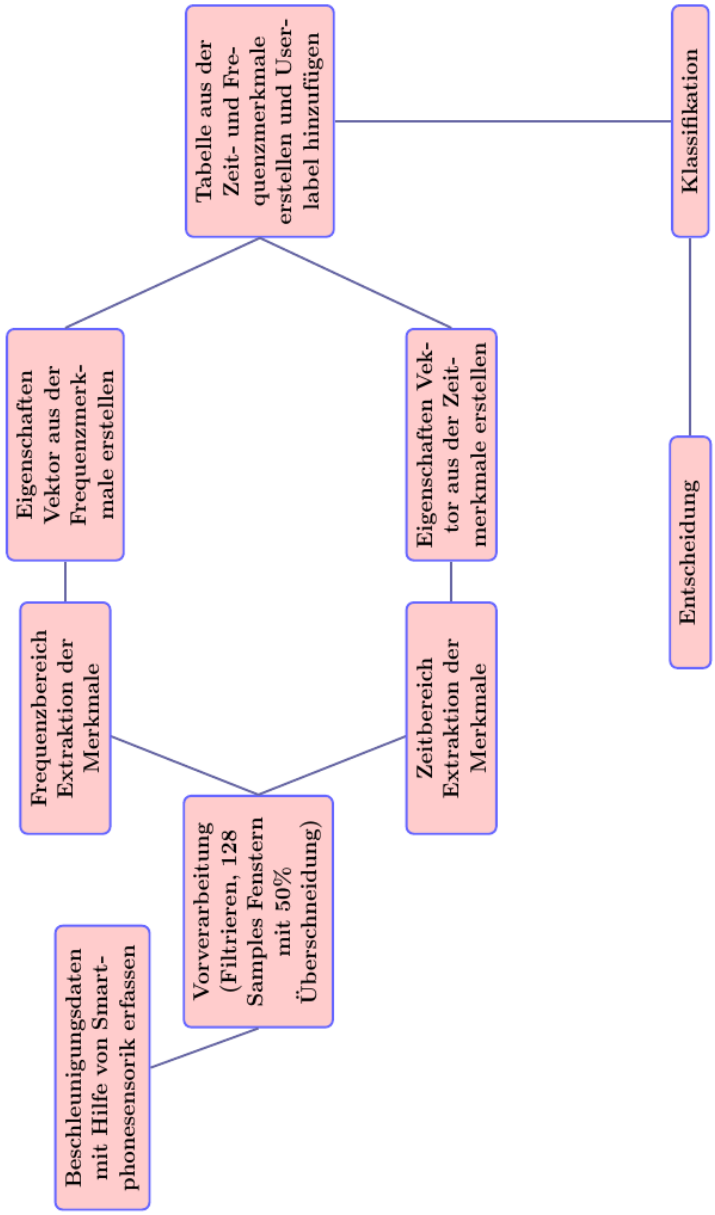

Abbildung 3.1: Blockdiagramm des Authentifizierungsverfahrens

3.2 Sensordatenerfassung

Für die Erfassung von Sensordaten (biometrische Merkmale der Benutzer) wird das Smartphone Galaxy S9 aus Abschnitt 2.1.1 verwendet. Hierfür wurde eine Software in Android Studio geschrieben. Eine Übersicht der Softwareoberfläche ist in Abbildung 3.2 dargestellt. Die Software funktioniert folgendermaßen:

- Der Kandidat tippt seinen Namen und sein Alter ein
- Mit Betätigen des On/Off-Schalters wird die Datenaufzeichnung gestartet
- Ab diesem Zeitpunkt müssen die Kandidaten 30 Sekunden in normalen Tempo gehen
- Die Aufzeichnung stoppt automatisch nach 30 Sekunden
- Mit Beendigung der Aufzeichnung wird eine `.csv`-Datei mit den biometrischen Samples generiert

Es wurden für Testzwecke Aufzeichnungen von 10 Kandidaten aufgenommen. Um ein breites Spektrum für die Untersuchung zu haben, wurden Kandidaten zwischen 11 und 79 Jahre ausgewählt, sowohl männlich als auch weiblich. Die Datenaufzeichnung erfolgt für jeden Kandidaten gleich. Eine Übersicht der Kandidaten ist in Tabelle 3.1 dargestellt.

Kandidat	**Geschlecht**	**Alter**
User 1	männlich	34
User 2	weiblich	26
User 3	männlich	29
User 4	weiblich	34
User 5	männlich	35
User 6	männlich	56
User 7	weiblich	54
User 8	weiblich	11
User 9	weiblich	73
User 10	männlich	79

Tabelle 3.1: Kandidaten Übersicht

Es wurden für jeden Kandidaten zwei Datensätze aufgezeichnet. Ein Datensatz besteht aus dreißig Sekunden Aufzeichnung. Jede Sekunde werden circa (ca.) 100 Samples aufgenommen. Das ergibt pro Datensatz 3000 biometrische Werte.

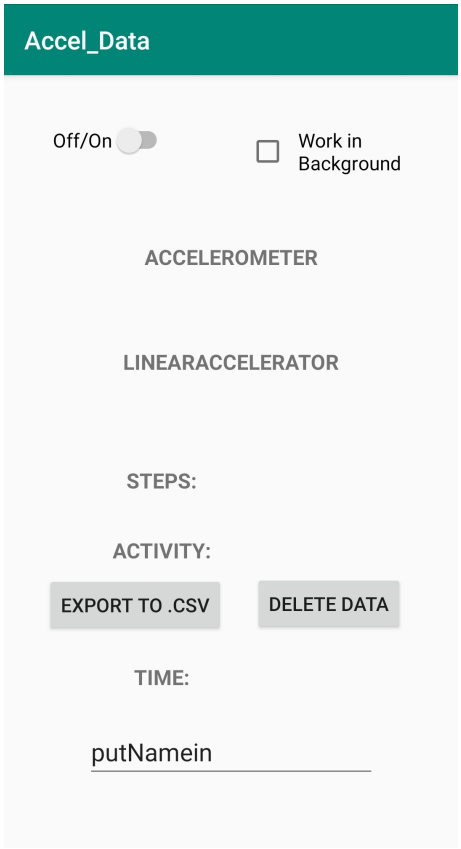

Abbildung 3.2: Erfassungssoftwareübersicht

Alle Kandidaten halten das Smartphone senkrecht in der Hand. Als Nachwirkung dieser Tatsache wird das Koordinatensystem aus 2.1 verdreht und die Y-Achse zeigt jetzt nach oben, die Bewegung findet entlang der Z-Achse und quer zur Bewegungsrichtung, der X-Achse, statt. Dies wird in der Abbildung veranschaulicht. Diese Erkenntnis ist relevant für die weitere Extraktion der Merkmale.

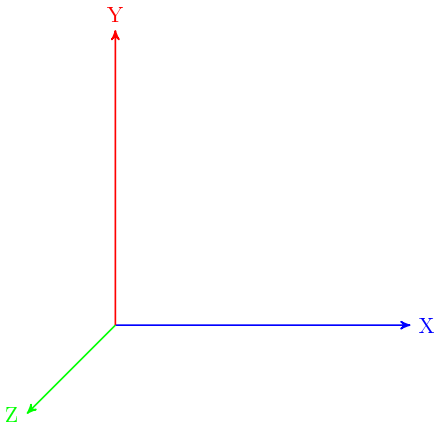

Abbildung 3.3: Koordinatensystem relativ zur Erde

3.3 Datenvorverarbeitung

Die Abbildung 3.4 stellt eine Perspektive über die Datenvorverarbeitung dar. Oben ist das Signal in Zeit- und unten in Frequenzbereich dargestellt. Es handelt sich in beiden Bildern über ein rohes und ein verarbeitetes Signal über eine Zeitspanne von ≈ 1.28 Sekunden. Die Fouriertransformation wird nach der Formel 3.1 durchgeführt:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-2\pi jnk}{N}} \tag{3.1}$$

X[k] - Diskreter Frequenzanteil an Stelle k
x[n] - Diskreter Zeitwert an Stelle n
N - Anzahl von Werten

Die genauere Vorgehensweise der Datenvorverarbeitung wird unten dargestellt.

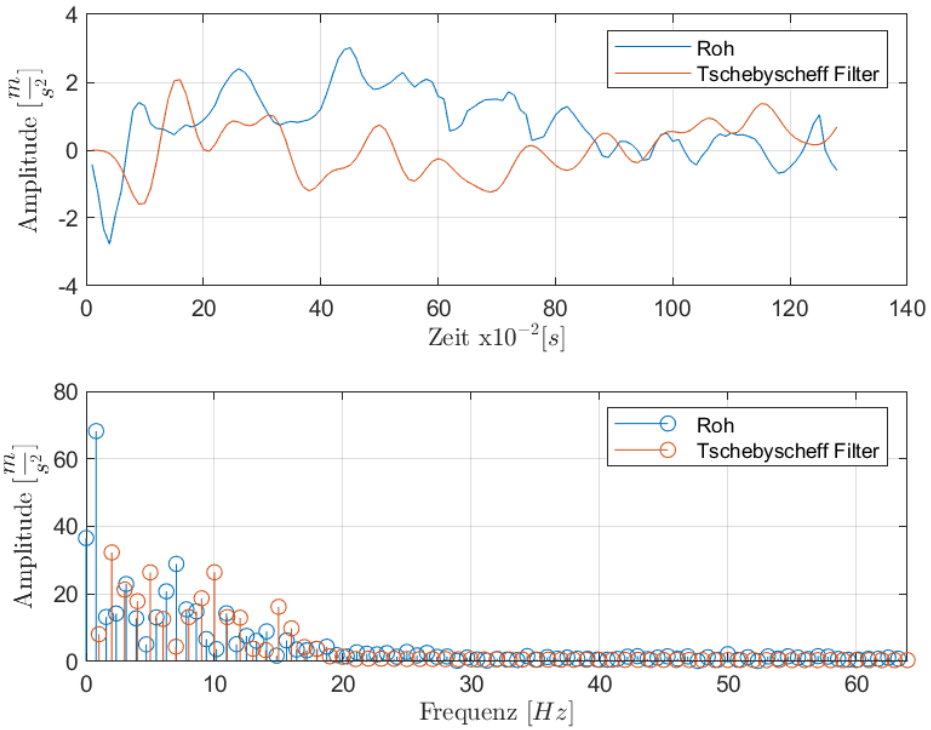

Abbildung 3.4: Datenvorverarbeitung Zeit- und Frequenzbereich

3.3.1 Filterung

Um unnötige Daten für die Weiterverarbeitung zu entfernen, wurden die Datensätze mit mehreren Filtern untersucht. Die besten Ergebnisse liefert ein Tschebyscheff-Filter mit Eckfrequenzen zwischen 0 und 12 Hz. Diese Grenzen sind auf das normale menschliche gehen abgestimmt, die in Abschnitt 2.2 untersucht wurden.

3.3.2 Normierung

Da die Daten vielleicht nicht immer in dem gleichen Bereich liegen und dieses ein Einfluss auf die Klassifizierung haben kann, werden die Daten auf dem Bereich [0,1] normiert. Bei der Modellierung hat eine Normierung der Daten nicht stattgefunden. Erst bei der Implementierung wurde diese Prozedur verwendet [15, S.141] .

3.3.3 Fehlende oder fehlerhafte Daten

Da manchmal Fehler bei der Datenerfassung auftreten können, kann es passieren, dass Daten fehlerhaft oder gar nicht aufgenommen werden. Diese Fehler können mehrere Ursachen haben zum Beispiel fehlerhafte Bauteile, Puffer oder Prozessorüberlastungen. Diese können auftreten, wenn während der Datenaufzeichnung Interrupts mit einer höheren Priorität aufgelöst werden. Wenn zu viele Fehler oder Nullwerte im Datensatz enthalten sind, dann werden die Datensätze verworfen und Neue aufgenommen. Wenn es sich nur um ein paar Werte handelt, dann können die vorhandenen Daten weiter verwertet werden.

3.3.4 Segmentierung

Für die Segmentierung werden die Roh-Datensätze im Fenster von 128 Sampels unterteilt, welches eine Zeit von ≈ 1.28 Sekunden entspricht. Der Wert für die Anzahl von Sampels im Fenster muss eine Zweierpotenz sein, ansonsten kann später bei der Implementierung keine Diskrete Fourier-Transformation (DFT) durchgeführt werden. Jedes Fenster hat eine 50% Überschneidung mit dem nächsten Fenster. Ausnahme hierfür sind das erste und letzte Fenster, welche jeweils nur 50% Werte enthalten. Die restlichen 50% werden mit Nullwerten gefüllt. Das ergibt aus einem Datensatz von 3000 Sampels, 47 Fenstern mit jeweils 128 Samples und eine Überschneidung von 50%. Diese Werte sind in einer MATLAB Matrix gespeichert. Jeder Kandidat erhält für die Datenverarbeitung in MATLAB eine 128 x 47 Matrix, bei der jede Spalte ein Fenster von 1.28 Sekunden entspricht.

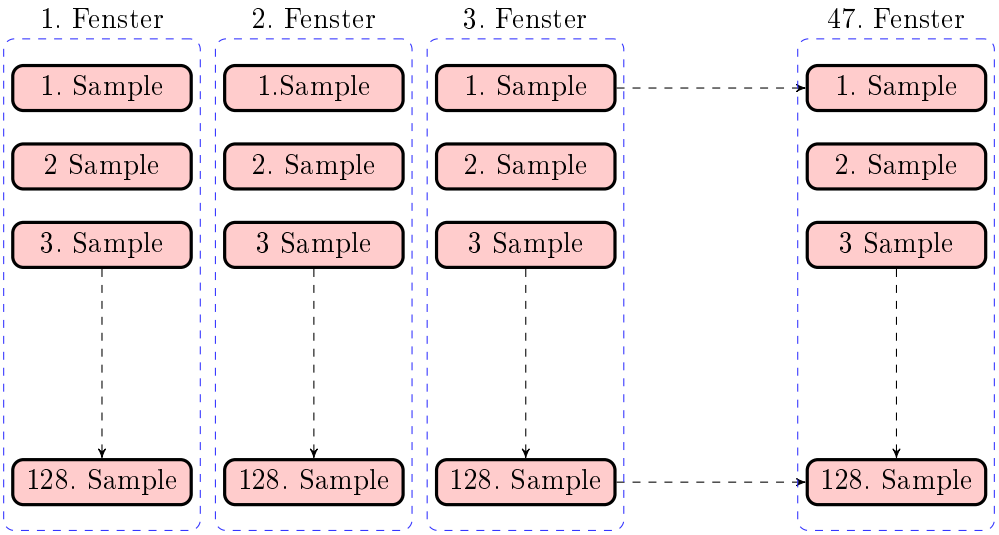

Abbildung 3.5: Übersicht der segmentierten Roh-Datei

Die Abbildung stellt den oben beschriebenen Prozess dar. Die Matrix wird in dieser Form für die Extraktion der Merkmale weitergereicht.

3.4 Extraktion der Merkmale

Aus den extrahierten Rohdaten werden verschiedene Merkmale generiert. Die Extraktion von biometrischen Merkmalen spielt für die biometrische Authentifizierung eine sehr wichtige Rolle. Anhand dieser Eigenschaften wird letztendlich entschieden, ob eine Person richtig oder falsch identifiziert wurde. Das Ziel besteht darin, eine Person anhand von Ähnlichkeiten oder Differenzen beim Gehen in den extrahierten Daten zu unterscheiden.

Die Extraktion von Merkmalen wird in zwei Klassen unterteilt. Es wird zuerst die Eigenschaft im Zeitbereich untersucht und dann im Frequenzbereich. Als Nächstes werden die verwendeten Funktionen gezeigt und anhand von Bildern und Formeln erläutert. Mit Hilfe des Datensatzes von User 1 werden jeweils für jede benutzte Funktion Bilder dargestellt. Die Werte entsprechen einer Aufzeichnung von 30 Sekunden. Es wird für jedes Merkmal ein Datensatz gemäß Abbildung 3.3 untersucht.

Hinweis In Abschnitt 3.4.1 und 3.4.2 beziehen sich die mit kleingeschriebenen Buchstabenparameter x, y, z auf die Zeitparameter-Werte, während die mit großgeschriebenen Buchstaben X, Y, Z sich auf Frequenzparameter beziehen.

3.4.1 Zeitbereich

Mittelwert

Mit Hilfe von Formel 3.2 wird für jeweils X-, Y-, und Z-Achse der Mittelwert berechnet. Dieser Prozess wird für alle 47 Fenster einzeln durchlaufen. Aus 128 Werten im Fenster ergibt sich dann ein Wert für den Mittelwert. Eine Ausgabe der Funktion ist in Abbildung 3.6 dargestellt.

$$\mu_{Zeit} = \frac{1}{N} \sum_{n=1}^{N} x[n] \tag{3.2}$$

μ_{Zeit} - Mittelwert für Zeit-Signal
x[n] - Zeit-diskrete Wert für X-, Y-, und Z-Achse
N - Anzahl von Werten

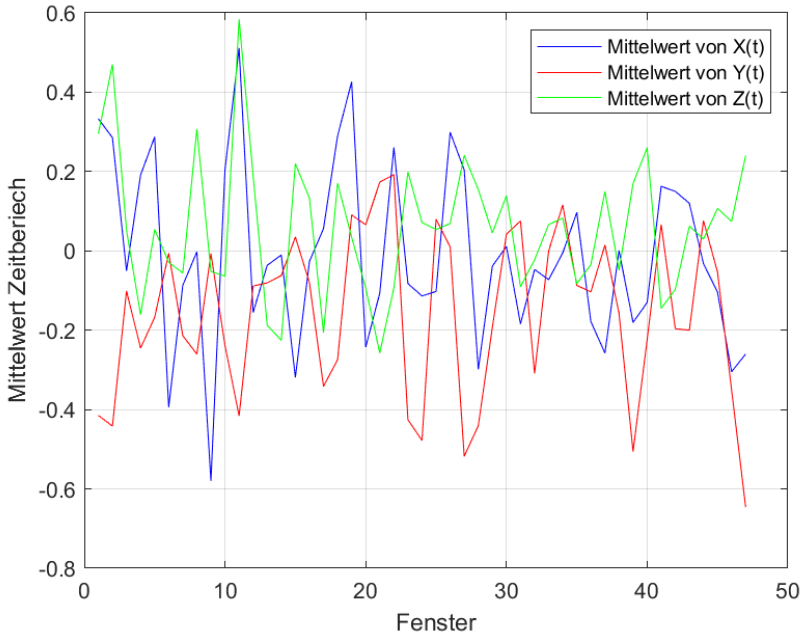

Abbildung 3.6: Zeitbereich: Mittelwert der Sensordaten

In Abbildung 3.6 ist eine sehr starke Änderung von Fenster zu Fenster zu beobachten. Die Werte alternieren abwechselnd um die Null-Linie. Daraus konnte ein Muster beobachtet werden. Ein Nachteil des Mittelwertes ist die Empfindlichkeit gegenüber Extremwerten.

Median

Als Nächstes wird der Median, auch Zentralwert genannt, berechnet. Auch hier ergibt sich ein Medianwert aus 128 Werten pro Fenster. Der Median ist der Wert in der Mitte einer Größe nach geordneter Datenreihe. Das heißt, mindestens 50% der Daten sind kleiner als der Median oder gleich dem Median und mindestens 50% der Daten sind größer als der Median oder gleich dem Median. Ein Vorteil dieser Funktion ist die Unempfindlichkeit gegenüber Ausreißer im Datensatz.

$$x_{Med,Zeit} = \frac{1}{2} \cdot (x_{\frac{N}{2}} + x_{\frac{N}{2}+1}) \tag{3.3}$$

$x_{Med,Zeit}$- Median für Zeit-Signal
N - Anzahl von Werten

Formel 3.3 bezieht sich nur auf Datensätze mit gerader Anzahl an Werten.

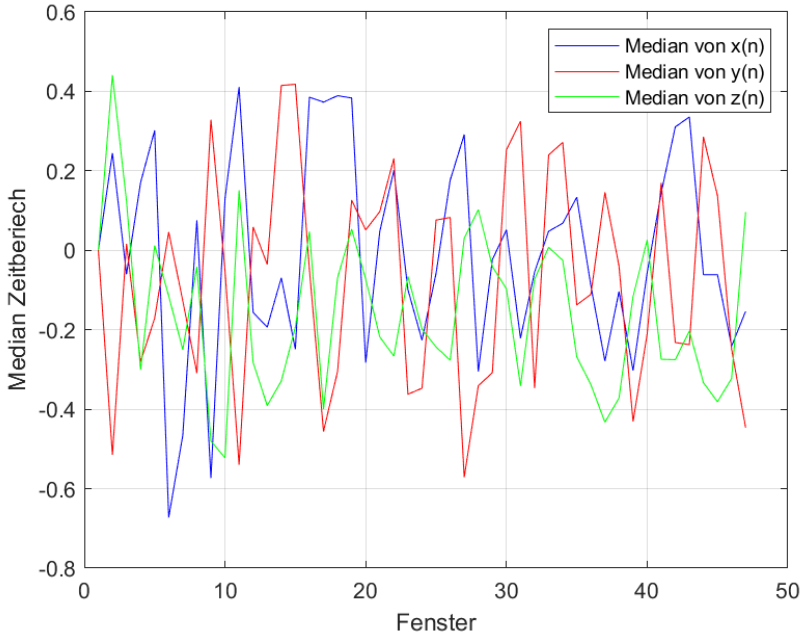

Abbildung 3.7: Zeitbereich: Median der Sensordaten

Abbildung 3.7 zeigt das Ergebnis der Medienfunktion. Auch in diesem Bild ist ein Muster zu erkennen.

Durchschnittliche Anzahl vom Peaks im Fenster

Mit Hilfe dieser Funktion lassen sich die Anzahl von Peaks in einem Fenster berechnen. Die Funktion untersucht die Anzahl von Peaks über die X-, Y- und Z-Achse. Als Ergebnis wird ein Mittelwert der drei Achsen gemäß Formel 3.2 durchgeführt.

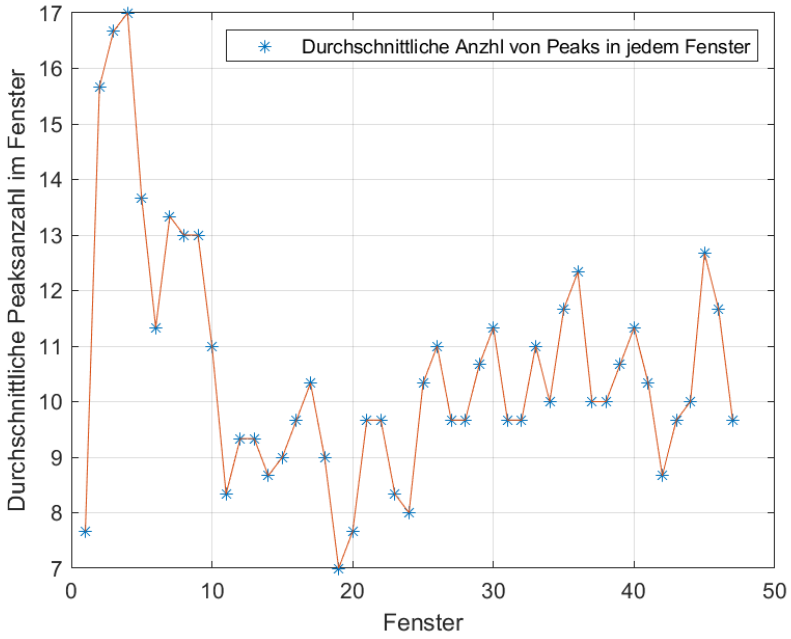

Abbildung 3.8: Zeitbereich: Durchschnittliche Anzahl von Peaks

Mit Hilfe dieses Merkmals kann ungefähr vorhergesagt werden, wie viel Schritte ein Kandidat durchläuft und wo ein Abschnitt in einem Schritt anfängt oder endet. Das Ergebnis der Funktion ist in Abbildung 3.8 dargestellt.

Durchschnittliche Distanz zwischen Peaks im Fenster

Diese Funktion gibt die Entfernung zwischen zwei aufeinanderfolgenden Ausreißern an. Die Ausgabe der Funktion wird als Durchschnittswert von Entfernungen zwischen zwei Peaks angegeben. Dieses Merkmal wird für jede Achse einzeln errechnet.

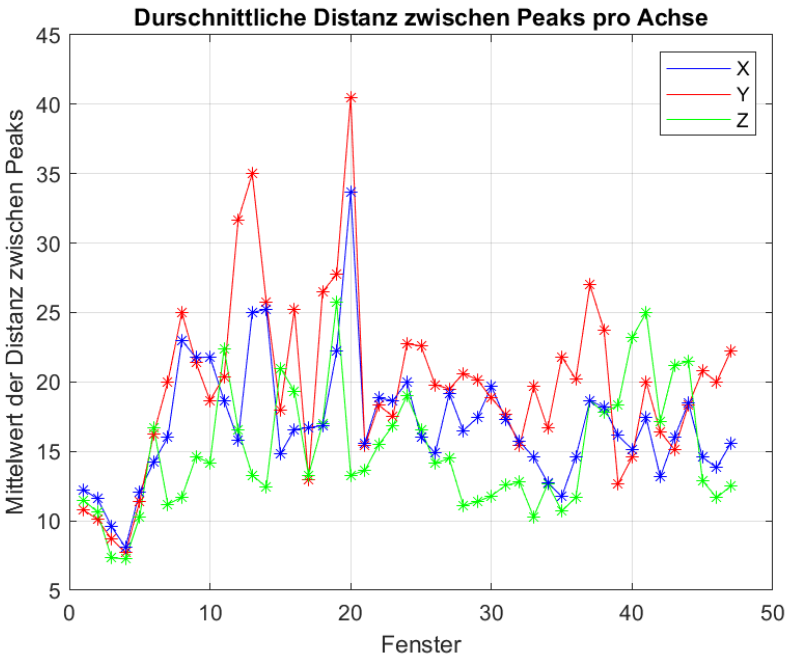

Abbildung 3.9: Zeitbereich: Durchschnittliche Distanz zwischen Peaks

Aus diesem Merkmal kann die Zeit zwischen den Schrittübergängen bestimmt werden. Es konnte auch etwas über die Schrittgeschwindigkeit besagen. Eine Übersicht dieses Merkmals ist in Abbildung 3.9 dargestellt.

Kreuzkorrelation

Für das nächste Merkmal wird die Korrelation zwischen jeweils X- und Y-Achse, X- und Z-Achse und anschließend Y-und Z-Achse berechnet (Abbildung 3.10). Diese Berechnung erfolgt mit Hilfe der folgenden Gleichungen:

$$Korr_{xy,Zeit} = \frac{\mu_x}{\mu_y} \tag{3.4}$$

$$Korr_{xz,Zeit} = \frac{\mu_x}{\mu_z} \tag{3.5}$$

$$Korr_{yz,Zeit} = \frac{\mu_y}{\mu_z} \tag{3.6}$$

[21, S.4 Formel (2) und (3)]

$Korr_{xy,Zeit}$ - Kreuzkorrelation zwischen x und y

$Korr_{xz,Zeit}$ - Kreuzkorrelation zwischen x und z

$Korr_{yz,Zeit}$ - Kreuzkorrelation zwischen y und z

μ_x- Mittelwert des X- Zeitwertes

μ_y- Mittelwert des Y- Zeitwertes

μ_z- Mittelwert des Z- Zeitwertes

Mit der Kreuzkorrelation wird der lineare Zusammenhang zwischen zwei Datensätzen deutlich. Mit anderen Worten wird die Ähnlichkeit zwischen zwei Achsen gemessen.

Abbildung 3.10: Zeitbereich: Korrelation

Quadratischer Mittelwert

Der quadratische Mittelwert oder auch als Root Mean Square (RMS) bekannt, ist derjenige Wert, der als Quadratwurzel des Quotienten aus der Summe der Quadrate für alle drei Achsen und ihre Anzahl an Werten berechnet ist. Der mathematische Ausdruck des quadratischen Mittelwertes wird durch die Formel 3.7 veranschaulicht.

$$QMW_{Zeit} = \sum_{i=1}^{N} \sqrt{\frac{x_i^2 + y_i^2 + z_i^2}{N}} \tag{3.7}$$

QMW_{Zeit} - Quadratischer Mittelwert
x_i - x-Wert an Stelle i
y_i - y-Wert an Stelle i
z_i - z-Wert an Stelle i
N - Anzahl von Werten

Der quadratische Mittelwert von einer periodischen Funktion ist gleich des QMW einer Periode dieser Funktion. Diese Eigenschaft des QMW kann etwas über die Periodizität besagen, in diesem Fall über die Wiederholung der Schritte.

Eine grafische Darstellung des QMW ist in Abbildung 3.11 dargestellt.

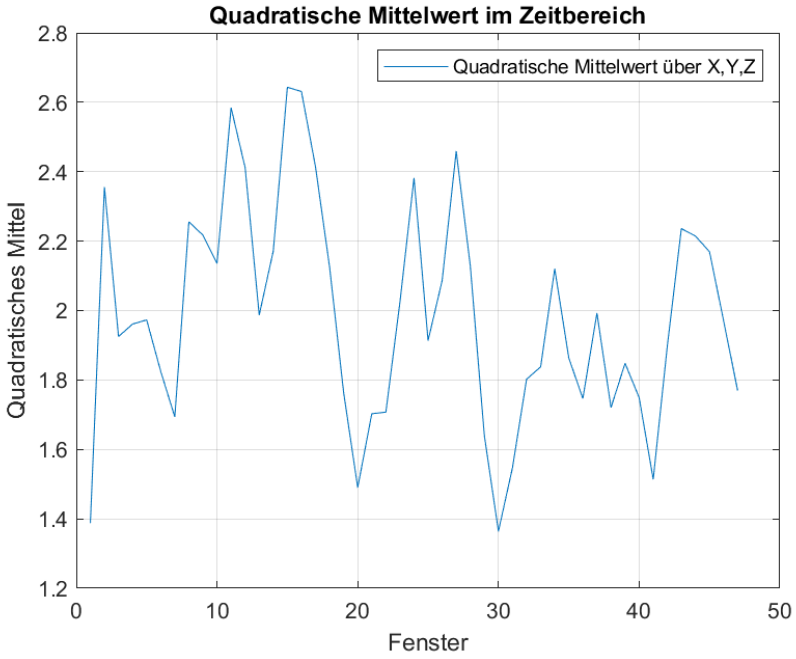

Abbildung 3.11: Zeitbereich: Quadratisches Mittel

Standardabweichung

Als Nächstes wird die Standardabweichung oder durchschnittliche Abweichung vom Mittelwert untersucht. Vereinfacht gesagt ist die Standardabweichung die durchschnittliche Entfernung aller gemessenen Ausprägungen eines Merkmals vom Durchschnitt. Auch hier wird die Standardabweichung für alle drei Achsen berechnet. Die Berechnung der Standardabweichung erfolgt über die Quadratwurzel der Varianz.

$$\sigma_{Zeit} = \sqrt{Var(x)} \tag{3.8}$$

$$Var(x) = E(x^2) - \mu_x^2 \tag{3.9}$$

σ_{Zeit} - Standardabweichung
$Var(x)$ - Varianz von x
$E(x)$- Erwartungswert von x
μ_x - Mittelwert von x

In dieser Untersuchung kann dieses Merkmal dafür benutzt werden, um Unterschiede beim Kandidaten festzustellen. Genauer gesagt kann etwas über die Körper- und Armschwingung vorhergesagt werden. Da dieser Durchlauf von Kandidat zu Kandidat unterschiedlich ist, kann daraus eine hilfreiche Erkenntnis gewonnen werden.

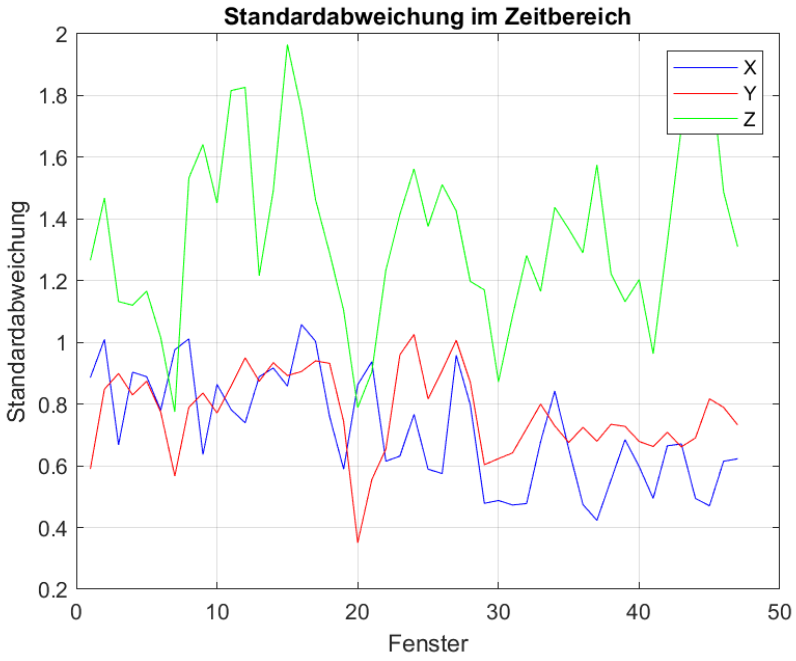

Abbildung 3.12: Zeitbereich: Standardabweichung

In Abbildung 3.12 wird die Standardabweichung des ersten Kandidaten dargestellt. Hier ist eine deutliche Abweichung vom Mittelwert in der Z-Achse zu sehen. Das ist auch zu erwarten, da die Z-Achse bei der Untersuchung der Achse entlang der Gangrichtung entspricht.

Maximalwert

Als Nächstes wir der Maximalwert untersucht. Diese Funktion, wie der Name schon sagt, sucht den Maximalwert in einem Fenster. Dieses Merkmal wird für alle drei Achsen angewendet.

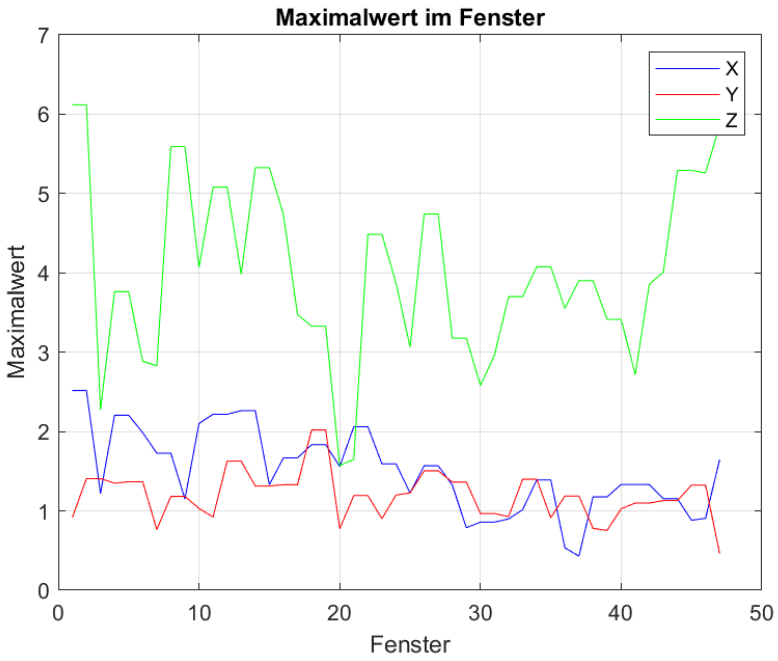

Abbildung 3.13: Zeitbereich: Maximalwert im Fenster

Diese Untersuchung liefert Informationen über zu hohe Werte in einem Fenster z. B. falsche oder fehlerhafte Werte. Die Abbildung 3.13 stellt den Maximalwert in einem Fenster für eine Periode von 30 Sekunden dar. Durch die 50% Datenüberschneidung zwischen zwei Fenstern kann beobachtet werden, dass meistens zwei benachbarte Werte gleich groß ausfallen.

Minimalwert

Analog zum Maximalwert wird an dieser Stelle das Minimum in einem Fenster untersucht. Auch hier erfolgt die Untersuchung entlang der X-, Y-, und Z-Achse. Hier liefert der Minimalwert eine Aussage über die Richtigkeit der Daten. Aus dem Minimalwert und dem Maximalwert kann auch die Gesamtgröße des Signals bestimmt werden.

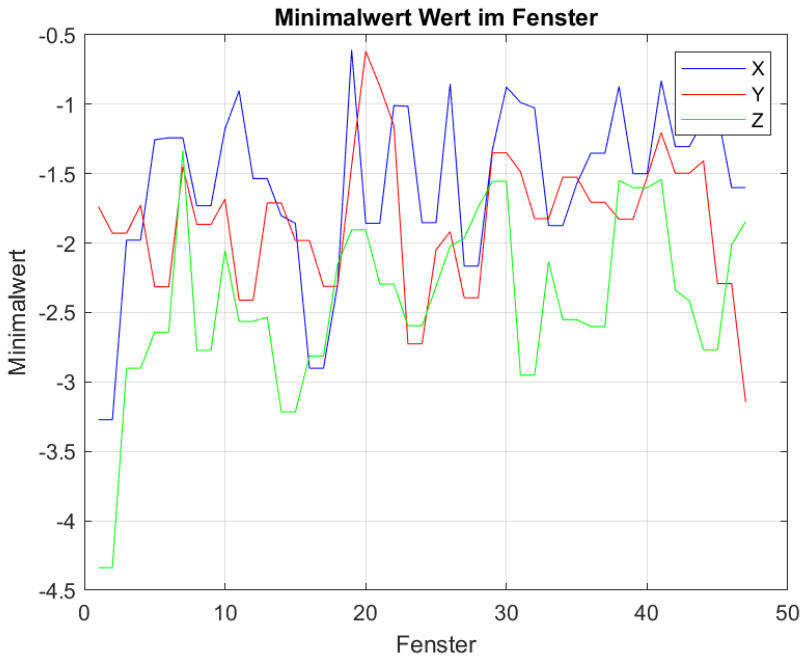

Abbildung 3.14: Zeitbereich: Minimalwert im Fenster

Die dargestellten Minimalwerte aus Abbildung 3.14 zeigen durch die 50% Überlappung einen ähnlichen Verlauf wie bei dem Maximalwert. Die Minimalwerte und Maximalwerte liefern nicht nur Angaben über die Richtigkeit der aufgenommenen Daten, sondern werden auch als Vergleich zwischen zwei Kandidaten benutzt.

Amplitude der ersten Werte

Dieses Merkmal soll dafür verwendet werden um einen Anfangspunkt zu definieren. Da es sich hier um ein wiederholtes Muster handelt, ist es wichtig einen Startpunkt definiert zu haben. Die Abbildung 3.15 stellt die Amplitude des ersten Wertes in jeweils dementsprechenden Fenstern dar.

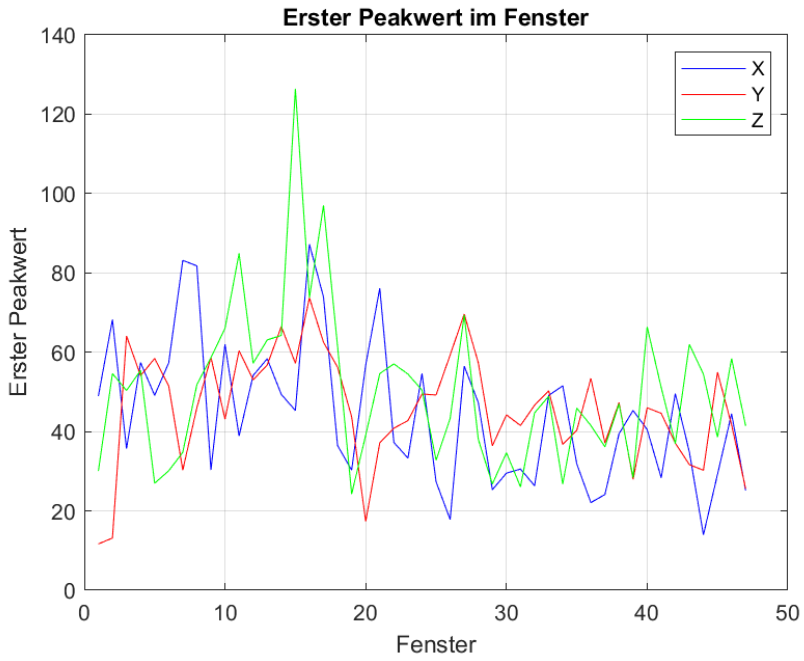

Abbildung 3.15: Zeitbereich: Amplitude der ersten Werte im Fenster

Amplitude der zweiten Werte

Für dieses Merkmal wird die Amplitude von dem zweiten Wert untersucht. Zusammen mit der Amplitude des ersten Wertes können wichtige Informationen über den Anfang des Signals gewonnen werden.

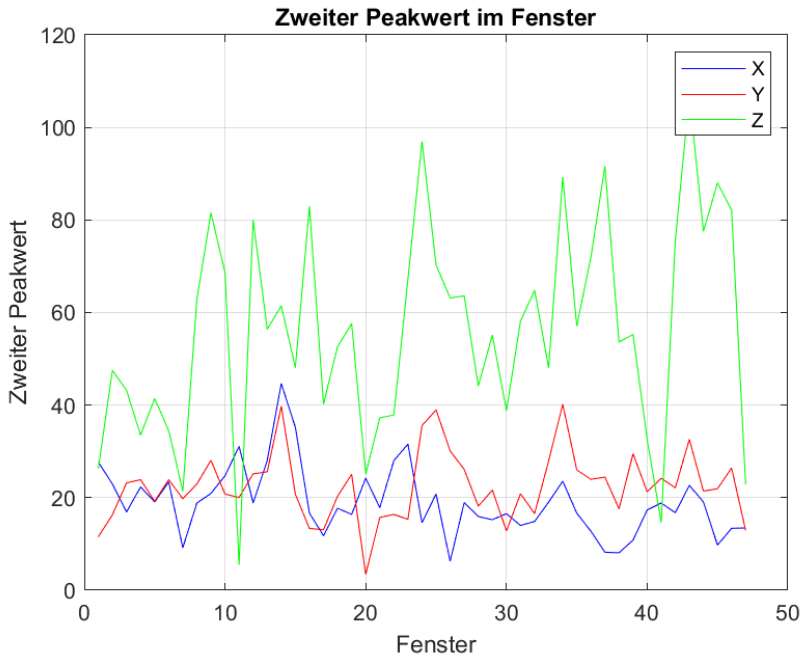

Abbildung 3.16: Zeitbereich: Amplitude der zweiten Werte im Fenster

Auch dieses Merkmal kann sehr gut als Unterscheidungsmerkmal zwischen zwei Kandidaten genutzt werden. Natürlich könnte mit diesem Muster auch der dritte-, vierte- und so weiter (usw.)- Wert in einem Fenster untersucht werden. Ähnliche Experimente zeigen aber, dass es ausreichend ist, nur den ersten und zweiten Amplitudenwert zu untersuchen.

3.4.2 Frequenzbereich

Die Frequenztransformation liefert eine zweite Sichtweise auf die biometrischen Sampels. Analog zum Zeitbereich werden hier die Merkmale der frequenztransformierten Signale untersucht. Zuerst wird das Zeitsignal gemäß Abbildung 3.5 mit Hilfe der Formel 3.1 Fourier transformiert.

Mittelwert

Das erste untersuchte Merkmal für die Frequenzwerte ist der Mittelwert.

$$\mu_{Frequenz} = \frac{1}{N} \sum_{k=1}^{N} X[k] \tag{3.10}$$

$\mu_{Frequenz}$ - Mittelwert des Frequenzspektrums
N - Anzahl von Werten
X[k] - Frequenz-diskreter Wert für X-, Y-, und Z-Achse an Stelle k

Es wird der Mittelwert für alle drei Achsen berechnet.

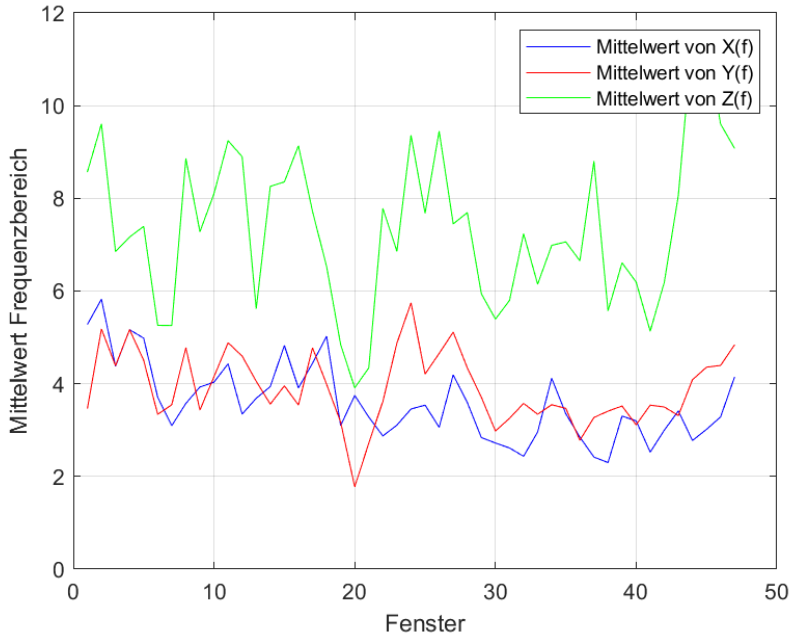

Abbildung 3.17: Frequenzbereich: Mittelwert der Sensordaten von Person 1

Der Mittelwert für den Frequenzbereich wird gemäß Formel 3.10 berechnet. Eine Ansicht des Mittelwertes der frequenztransformierten Werte wird in Abbildung 3.17 dargestellt.

Median

Zunächst wird der Median des Frequenzspektrums bestimmt. Hierfür wird die Formel 3.11 verwendet.

$$X_{Med,Frequenz} = \frac{1}{2} \cdot (X_{\frac{N}{2}} + X_{\frac{N}{2}+1}) \tag{3.11}$$

$X_{Med,Frequenz}$ - Medianwert des Frequenzspektrums
N - Anzahl von Werten

Die Formel 3.11 bezieht sich nur auf Datensätze mit gerader Anzahl an Werten. Die Abbildung 3.18 zeigt die Medianwerte für das Frequenzspektrum.

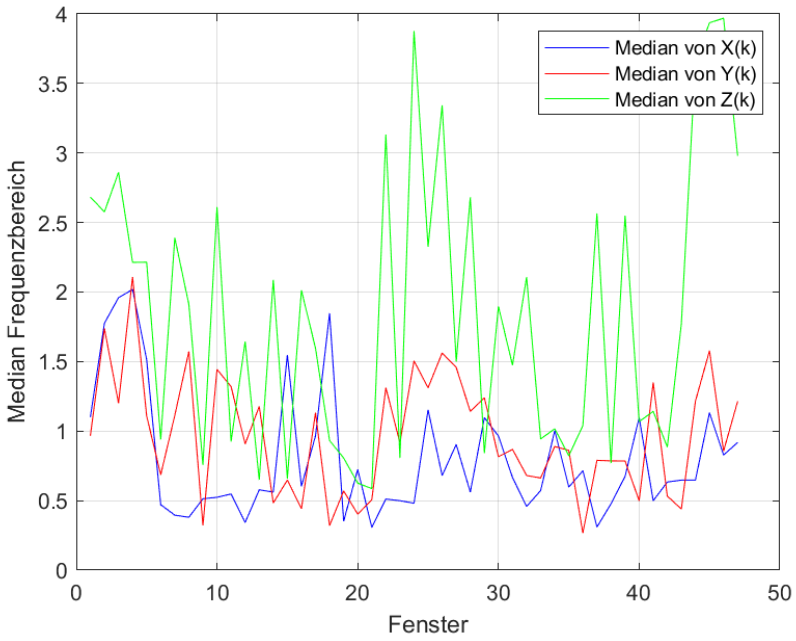

Abbildung 3.18: Frequenzbereich: Median der Sensordaten

Schwerpunktwellenlänge

Die Schwerpunktwellenlänge, auch bekannt als (en. Spectral Centroid), gibt an, wo sich der Mittelpunkt des Spektrums befindet. Der Spektralschwerpunkt kann als der erwar-

tete Wert der Spektralverteilung eines Spektrums angesehen werden. Größere Werte des Schwerpunktes reflektieren eine Tendenz zu höheren Frequenzen, während kleinere Werte die niedrigeren Frequenzen reflektieren.

$$\lambda_C = \sum_{k=1}^{N} \frac{x[k] \cdot X[k]}{N} \tag{3.12}$$

[21, S.4 Formel(4)]
λ_C - Schwerpunktwellenlänge
x[n] - Zeit-diskreter Wert
X[k] - Frequenz-diskreter Wert
N - Anzahl von Werten

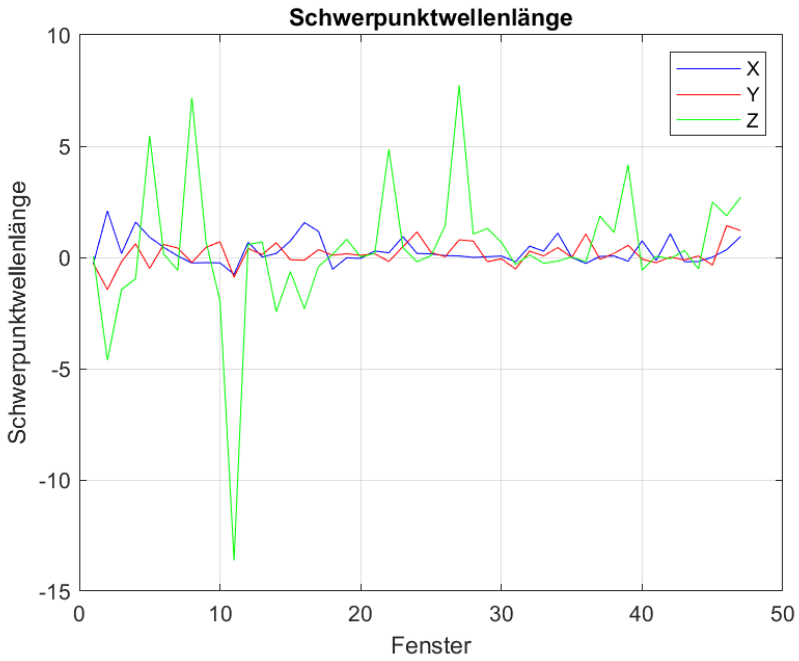

Abbildung 3.19: Frequenzbereich: Median der Sensordaten

Der Spektralschwerpunkt für die X-, Y-, und Z-Achse wird in Abbildung 3.19 dargestellt.

Kreuzkorrelation

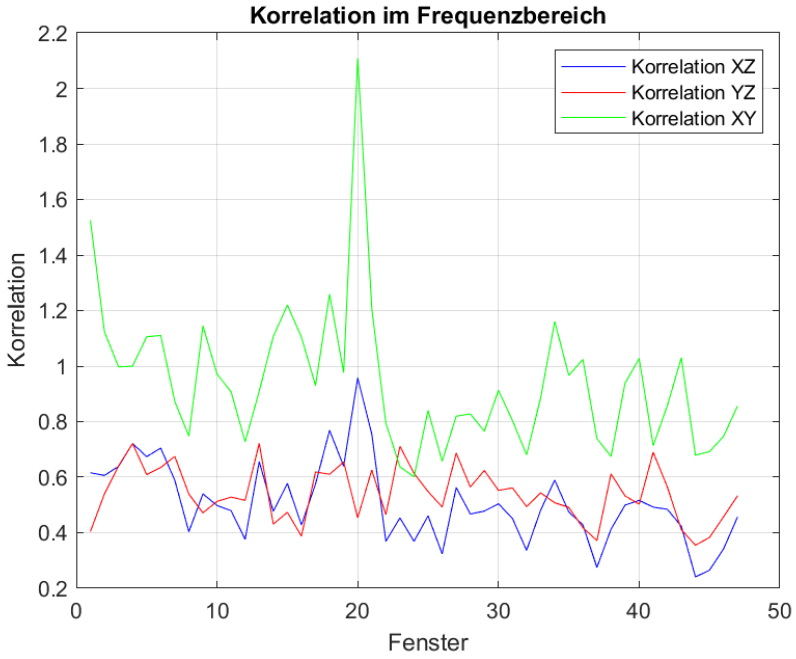

Abbildung 3.20: Frequenzbereich: Korrelation

$$Korr_{XY,Frequenz} = \frac{\mu_X}{\mu_Y} \tag{3.13}$$

$$Korr_{XZ,Frequenz} = \frac{\mu_X}{\mu_Z} \tag{3.14}$$

$$Korr_{YZ,Frequenz} = \frac{\mu_Y}{\mu_Z} \tag{3.15}$$

[21, S.4 Formel (2) und (3)]

$Korr_{XY,Frequenz}$ - Kreuzkorrelation zwischen X und Y

$Korr_{XZ,Frequenz}$ - Kreuzkorrelation zwischen X und Z

$Korr_{YZ,Frequenz}$ - Kreuzkorrelation zwischen Y und Z

μ_X- Mittelwert der X- Frequenzwerte
μ_Y- Mittelwert der Y- Frequenzwerte
μ_Z- Mittelwert der Z- Frequenzwerte

Die Abbildung 3.20 zeigt den linearen Zusammenhang zwischen X und Y, X und Z und Y und Z Frequenzspektren.

Quadratischer Mittelwert

Der quadratische Mittelwert im Frequenzbereich wird analog wie bei der Zeitberichtfunktion berechnet. Die Formel 3.16 zeigt die mathematische Darstellung des quadratischen Mittelwertes im Frequenzbereich.

$$QMW_{Frequenz} = \sum_{i=1}^{N} \sqrt{\frac{X_i^2 + Y_i^2 + Z_i^2}{N}} \tag{3.16}$$

$QMW_{Frequenz}$- Quadratischer Mittelwert
X_i - X-Wert an Stelle i
Y_i - Y-Wert an Stelle i
Z_i - Z-Wert an Stelle i
N - Anzahl von Werten

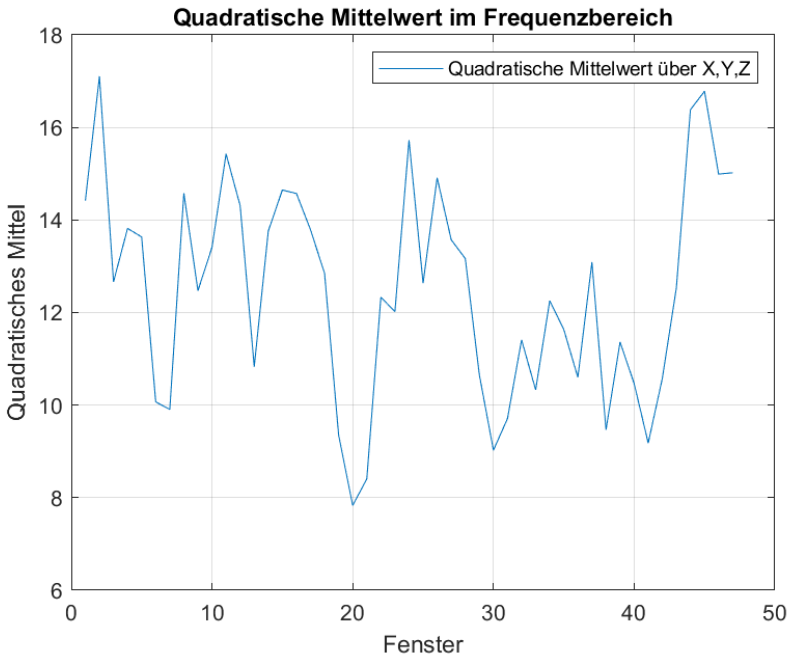

Abbildung 3.21: Frequenzbereich: Quadratisches Mittel

Die Abbildung 3.21 zeigt den quadratischen Mittelwert der Frequenzachsen X, Y, Z.

Standardabweichung

Die Standardabweichung für das Frequenzspektrum wird mit Hilfe der Formel 3.17 berechnet.

$$\sigma_{Frequenz} = \sqrt{Var(X)} \tag{3.17}$$

$$Var(X) = E(X^2) - \mu_X^2 \tag{3.18}$$

$\sigma_{Frequenz}$ - Standardabweichung
$Var(X)$ - Varianz von X
$E(X)$- Erwartungswert von X
μ_X - Mittelwert von X

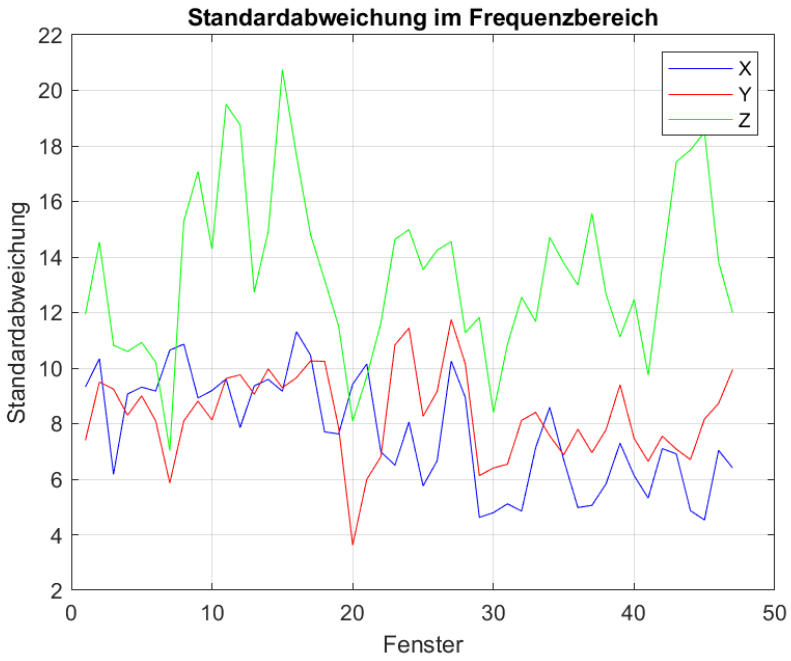

Abbildung 3.22: Frequenzbereich: Standardabweichung

Die Abbildung 3.22 zeigt die Abweichung von dem Frequenzmittelwert für die jeweilige X-, Y-, und Z-Achse.

Tabelle 3.2 zeigt eine Übersicht der oben beschriebenen Funktionen. In der Spalte *Symbol* wird anstelle des Buchstaben (a) - x, y und z und für den Buchstaben (A) - X, Y und Z eingesetzt.

	Feature Typ		Symbol	Anzahl Vektoren
1	**Zeit**	Mittelwert	μ_{Zeit}	3
2		Median	$a_{Med,Zeit}$	3
3		Durchschnittliche Anzahl vom Peaks im Fenster	DAP	1
4		Durchschnittliche Distanz zwischen Peaks im Fenster	DDP	3
5		Kreuzkorrelation	$Korr_{aa,Zeit}$	3
6		Quadratischer Mittelwert	QMW_{Zeit}	1
7		Standardabweichung	σ_{Zeit}	3
8		Maximalwert	Max_A	3
9		Minimalwert	Min_A	3
10		Amplitude der ersten Werte	$A1$	3
11		Amplitude der zweiten Werte	$A2$	3
12	**Frequenz**	Mittelwert	$\mu_{Frequenz}$	3
13		Median	$A_{Med,Frequenz}$	3
14		Schwerpunktwellenlänge	λ_C	3
15		Kreuzkorrelation	$Korr_{AA,Frequenz}$	3
16		Quadratischer Mittelwert	$QMW_{Frequenz}$	1
17		Standardabweichung	$\sigma_{Frequenz}$	3
Gesamtanzahl				35

Tabelle 3.2: Merkmaltabelle

Template

An dieser Stelle wird mit Hilfe der extrahierten Funktionen eine Merkmaltabelle, auch bekannt als Feature-Tabelle, erstellt. Für jedes extrahierte Merkmal wird ein Vektor mit dessen Werten erstellt. Diese Merkmal-Vektoren werden die Spalten der Tabelle bilden und sind mit blau in Abbildung 3.23 gekennzeichnet. Die Reihen der Tabelle entsprechen sogenannten Instanzen und werden mit rot in der Abbildung 3.23 gekennzeichnet. Eine Instanz entspricht einem Fenster von etwa 1.28 Sekunden mit einer 50% Überschneidung des nächsten Fensters.

Die letzte Spalte der Tabelle, in diesem Fall die 36. Spalte, wird dann mit einem Label

gekennzeichnet. Dieses Label beschreibt den Kandidaten, von dem die biometrischen Merkmale extrahiert wurden. Der Prozess wird in dieser Arbeit als Labeling bezeichnet. Eine graphische Darstellung von dem Aufbau eines Templates wird in der Abbildung 3.23 dargestellt.

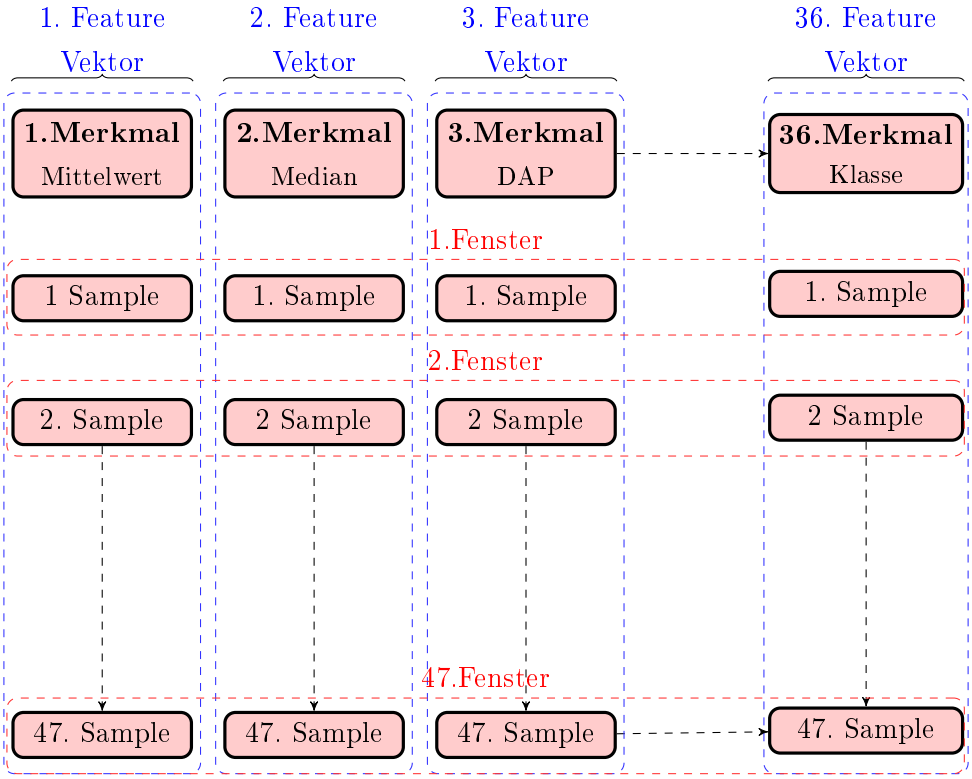

Abbildung 3.23: Aufbau der Feature-Template

3.5 Klassifizierung

Für diesen Versuch werden Klassifikatoren (en. Clasifier) auf Basis von maschinellem Lernen eingesetzt. Es wird zwischen drei Klassen von Klassifikatoren unterschieden:

- Klassifikatoren basierend auf der Entscheidungstheorie.
- Lineare Klassifikatoren.
- Nicht-lineare Klassifikatoren.

Ein maschinelles Lernsystem versucht die Zusammenhänge zwischen den Daten und den gewünschten Ergebnissen zu erkennen beziehungsweise zu erlernen, um als Ergebnis ein sogenanntes Modell zu liefern. Dieser Prozess geschieht meist in mehreren Schritten:

- Ein Modell wird erstellt (trainiert)
- Das Modell wird untersucht (getestet)
- Das Modell wird verbessert

Diese Abfolge wiederholt sich, bis das Ergebnis zufriedenstellend ist. Ist ein Modell bereits trainiert, kann es anhand einer gegebenen Eingabe ein Ergebnis bzw. eine Vorhersage liefern.

Für die vorgestellte Problemdarstellung in dieser Arbeit wird eine Zwei-Klassen Klassifizierung verwendet. Das bedeutet für die Ausgabe der Klassifizierung, dass die Daten in zwei Klassen unterteilt werden [4, S.180-185].

Es wird eine eins-gegen-eins (en. one on one) Untersuchung stattfinden. Der echte Kandidat wird mit seinem Gegner verglichen.

Bei einer Problemdarstellung mit mehr als zwei Klassen handelt es sich um Multi-Klassen Klassifikationen.

Als Erstes wird die Feature-Tabelle gemäß Abbildung 3.23 in Lern- (en.Train) und Testdaten untergliedert. Diese Unterteilung erfolgt in der Regel 60% für Lerndaten und 40% für Testdaten. Diese Parameter können natürlich variiert werden. Obwohl dieser Prozess zuverlässig ist, kann nicht mit hundertprozentiger Sicherheit eine Aussage für die ganzen Sampels gegeben werden.

Das Problem liegt hier in der Tatsache, dass die Testdaten nie trainiert werden und damit ein gewisser Prozentsatz an Genauigkeit verloren geht.

Für eine bessere Beurteilung der Vorhersagegenauigkeit sollte die dargestellte Klassifizierung mehrfach mit wechselnden Lern- und Testdaten durchgeführt werden.

Um dieses Problem zu umgehen wird in dieser Arbeit ein sogenanntes k-fache Kreuzvalidierungsverfahren (en. k-Fold Cross validation) verwendet [7, S.483].

Die Feature-Tabelle wird hier in k-disjunkte Teilmengen partitioniert. Jede von den Teilmengen dient daraufhin reihum als Teststichprobe, während die jeweils verbleibenden (k - 1) Stichproben gemeinsam die Trainingsstichprobe bilden.

Der Mittelwert der Verlustfunktion über die k-te Wiederholungen dient dazu, die Generalisierbarbeit der Parameterschätzung insgesamt zu beurteilen.

Es wurde für diese Arbeit für k der Wert 10 ausgewählt.

Bei einem maschinellen Lern-Problem ist es sinnvoll gleich mehrere Klassifikatoren auszuprobieren. Zunächst wird eine Übersicht der Klassifikatoren präsentiert.

3.5.1 Support Vector Machine

SVM Klassifikatoren sind sehr gut für die Lösung von Zwei-Klassen bzw. binären Problemen geeignet. Die generelle Idee hinter der SVM Implementierung ist, die Daten in einer höheren Dimension (Hyperebene H) darzustellen, bei der eine lineare Unterscheidung möglich ist.

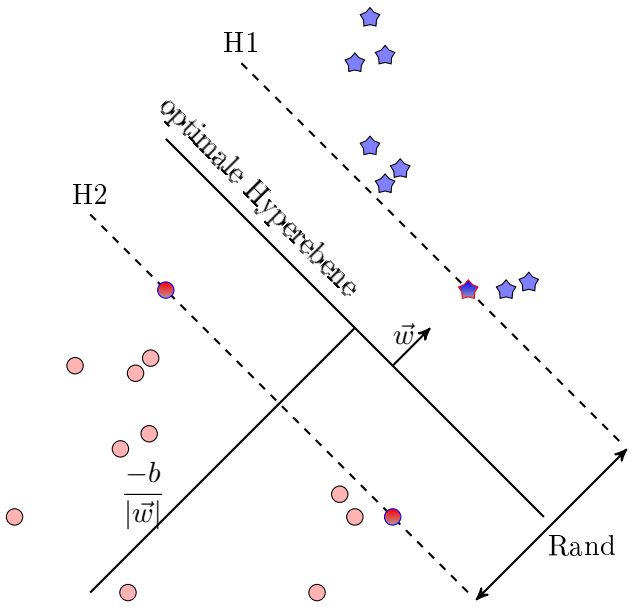

Abbildung 3.24: SVM: Hyperebenen für Trennung von zwei Klassen

Die Abbildung 3.24 zeigt die Zwei-Klassen getrennt von der optimalen Hyperebene. Die Support-Vektoren sind mit dunkler Farbe gekennzeichnet.

Zuerst wird die lineare Ebene untersucht. Bei der Untersuchung wird die separierende Ebene mit

$$H = \{\vec{x} | \langle \vec{w}, \vec{x} \rangle + b = 0\} \tag{3.19}$$

definiert. $\langle \vec{w}, \vec{x} \rangle$ ist das Vektorprodukt zwischen $\vec{w}$ und $\vec{x}$.

Der Abstand zur Ebene ist folgendermaßen definiert:

$$dist(\vec{x}, H) = \left| \frac{1}{\|\vec{w}\|} (\langle \vec{w}, \vec{x} \rangle + b = 0) \right| \tag{3.20}$$

Da die Daten in Zwei-Klassen klassifiziert sind, wird Folgendes mathematisch beschrieben:

$$y_i = sgn(\langle \vec{w}, \vec{x} \rangle + b) \tag{3.21}$$

$$y_i \; \epsilon \; \{\text{-1,1}\}$$

Als Nächstes muss der Rand beschrieben werden. Dafür muss der minimale Abstand $dist(\vec{x}, H)$ so groß wie möglich werden (Maximierungsproblem):

$$\xi = min \left| \frac{1}{\|\vec{w}\|} (\langle \vec{w}, \vec{x} \rangle + b = 0) \right| \tag{3.22}$$

Für die Feststellung der Hyperebene mit dem höchsten Abstand wird das beschränkte Optimierungsproblem verwendet:

$$min \frac{1}{2} \|\vec{w}\|^2 \tag{3.23}$$

Um dieses Problem zu lösen, wird die Lagrangefunktion eingeführt. Durch die Einführung von Lagrangemultiplikatoren $\alpha_i > 0$ kann die Beschränkung in die Funktion mit einfließen.

$$L(\vec{w}, b, \alpha) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle \vec{w}, \vec{x} \rangle + b) - 1) \tag{3.24}$$

[4, S.328 Formel (7.7)]

Die Funktion kann maximiert werden, indem man sie nach b und $\vec{w}$ ableitet.

Daraus folgt:

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} \alpha_i \alpha_m y_i y_m \langle \vec{x}_i, \vec{x}_m \rangle \tag{3.25}$$

[4, S.329 Formel (7.10)]

Dieses Problem kann mit verschiedenen Algorithmen aus der Optimierungstheorie gelöst werden.

An dieser Stelle wird das Kernelkonzept eingeführt. Wenn es nicht möglich ist, die Daten direkt linear zu trennen, dann werden diese mit einer Hilfsfunktion Φ in einen neuen Raum mit höherer Dimension dargestellt um zum Schluss wesentlich einfachere, lineare Formeln darauf anzuwenden und somit optimal zu trennen. Das bedeutet nur die Ersetzung des Skalarproduktes in die Formel 3.25 mit der Hilfsfunktion Φ.

$$\langle \vec{x}_i, \vec{x}_m \rangle \longrightarrow \langle \Phi(\vec{x}_i), \Phi(\vec{x}_m) \rangle \tag{3.26}$$

Es wird ab jetzt anstelle des Skalarproduktes eine Funktion K_Φ, die folgendermaßen definiert ist, betrachtet:

$$K_\Phi(\vec{x}, \vec{y}) = \langle \Phi(\vec{x}), \Phi(\vec{y}) \rangle \tag{3.27}$$

Anhand voreingestellter K_Φ Funktionen ist es möglich, einer Reihe von Optimierungen für die SVM Klassifikation durchzuführen [4, S.259-265].

Ein paar Beispiele für sehr häufig genutzte Kernelfunktionen sind im Folgenden aufgelistet:

- lineare Kernelfunktion

$$K(\vec{x}, \vec{y}) := \langle \vec{x}, \vec{y} \rangle \tag{3.28}$$

- Radial-Basis-Kernelfunktion (Gaussian)

$$K(\vec{x}, \vec{y}) := exp(-\gamma \cdot |\vec{x} - \vec{y}|^2) \tag{3.29}$$

- Polynomiale-Kernelfunktion

$$K(\vec{x}, \vec{y}) := (\langle \vec{x}, \vec{y} \rangle + 1)^d \tag{3.30}$$

- Sigmoid-Kernelfunktion

$$K(\vec{x}, \vec{y}) := tanh(\gamma \cdot (\vec{x} - \vec{y}) + c) \tag{3.31}$$

3.5.2 Die k-nächste-Nachbarn-Klassifikation

Die k-nächste-Nachbarn (en. k-Nearest Neighbor) (k-NN) ist ein einfacher Instanz-basierender Algorithmus. Auch für diesen besteht die gespeicherte Templatetabelle aus Instanzen mit dem zugehörigen User-spezifischen biometrischem Merkmal. Jede Instanz wird mit dem zugehörigen User-Label gekennzeichnet.

Während der Klassifikation wird der Euklidische Abstand aus Formel 3.32 zwischen dem geprüften und dem gespeicherten Kandidaten untersucht.

$$Euk(r, p) = \sqrt{\sum_{i=1}^{N} (q(i) - p(i))^2} \tag{3.32}$$

Die Label-Klasse von den k-Vektoren mit der niedrigsten Distanz werden untersucht. Es wird nach dem Mehrheitsprinzip gehandelt. Die Klasse, die am meisten vertreten ist,

wird den untersuchten Instanzen zugeordnet. Falls es keine Mehrheit gibt, werden die untersuchten Instanzen mit der gespeicherten Klasse gekennzeichnet (es handelt sich um den echten Kandidaten).

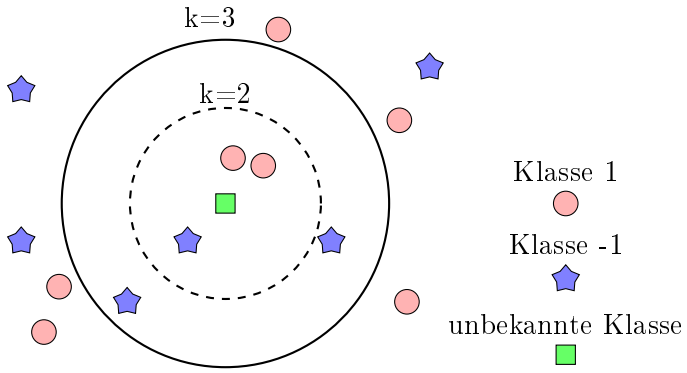

Abbildung 3.25: k-NN: zwei Klassen Klassifizierungsverfahren

Die Abbildung 3.25 stellt die k-nächste-Nachbarn-Klassifikation dar. Für ein k=2 wird für die unbekannte Klasse die Distanz zu den nächsten zwei gleichen Nachbarn aus einer Klasse berechnet. In diesem Beispiel wird das grüne Rechteck der Klasse mit der Klasse des roten Kreises gekennzeichnet. Für k=3, die Distanz zu den nächsten drei gleichen Nachbarn aus einer Klasse, bekommt das grüne Rechteck die Klasse des blauen Sterns.

3.6 Entscheidung

Unabhängig davon welches Klassifikationsmodell verwendet wird, können die Ergebnisse der Klassifikation weiter verarbeitet werden. Die Entscheidung basiert auf dem Ergebnis der einzelnen untersuchten Instanzen.

Jede Instanz bekommt ein Label, gehört entweder dem gespeicherten Label (positiv) oder nicht (negativ) dazu. Am Ende werden die positiv gelabelten Instanzen gezählt. Danach wird ein Quotient zwischen positiv gelabelten Instanzen und den gesamten untersuchten Instanzen berechnet.

Die Entscheidung wird anhand einer voreingestellten Grenze (Threshhold) ausfallen. Wenn diese Grenze erreicht wird, dann würde der Kandidat als positiv authentifiziert werden. Wenn dieser Wert unter dieser Grenze liegt, dann handelt es sich um eine nicht-erfolgreiche Authentifizierung.

3.7 Auswertung der Ergebnisse

In diesem Abschnitt werden die Ergebnisse des MATLAB-Modells anhand der aufgezeichneten biometrischen Daten der Kandidaten vorgestellt. Als Erstes werden nur die Ergebnisse von User 1 und User 2 im Detail untersucht. Als Zweites wird eine Gesamtübersicht der Ergebnisse für alle Kandidaten präsentiert. Anschließend wird eine Parameterauswahl für die Implementierung anhand der ausgewerteten Ergebnisse getroffen.

3.7.1 Detaillierte Untersuchung der Ergebnisse zwischen User 1 und User 2

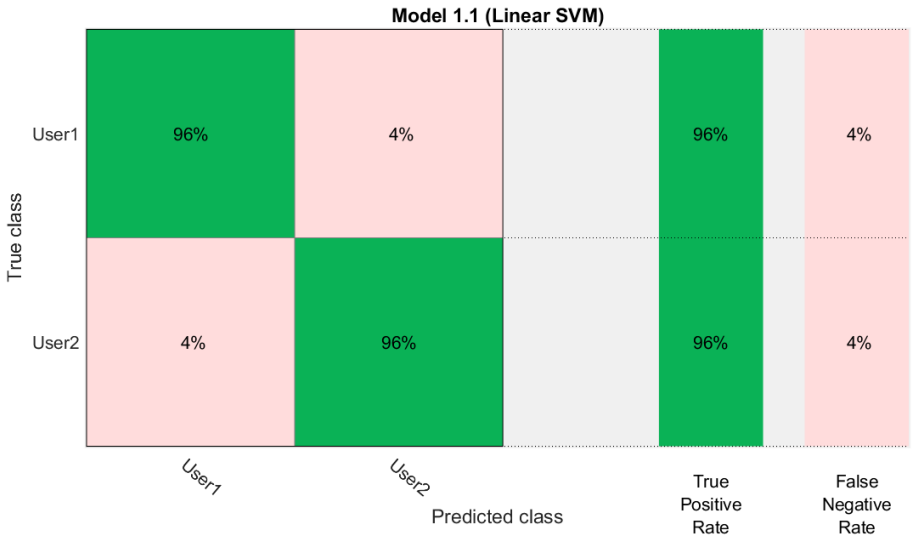

Abbildung 3.26: Plot der Confusions Matrix (dt. Klassifikationstabelle) zwischen User 1 und User 2

Die Abbildung 3.26 stellt die Ergebnisse des linearen SVM Klassifikators als Klassifikationstabelle (en. Confusions Matrix) dar. Die Y-Achse stellt die echten Klassen dar, während die X-Achse die Vorhersage der klassifizierten Klassen darstellt. Der Klassifikator wurde mit einem 10-fachen Kreuzvalidierungsverfahren untersucht. Die Diagonale der Matrix in Abbildung 3.26 (linke Seite), mit grün gekennzeichnet, zeigt die richtigen klassifizierten Ergebnisse (en. True Positive Rate)(TPR) in Prozent für jeweils User 1 im ersten Feld und User 2 im zweiten Feld an. Die anderen zwei Felder, mit rosa gekennzeichnet, stellen die falschen klassifizierten Ergebnisse (en.False Negative Rate(FPR)) für

jeweils User 1 in der ersten Reihe und User 2 in der zweite Reihe dar. Die rosa Felder stellen die Gleichfehlerraten, präsentiert im Abschnitt 2.3.4, dar. Oben ist die FRR und unten die FAR dargestellt. Für diesen Fall wird für den EER ein Wert von 0.045 erreicht. Diese Fehlerrate ist sehr niedrig und spricht somit für einen sehr guten Klassifikator.

Konkret wird die Confusions Matrix folgendermaßen interpretiert: für 96% Prozent der Fälle konnte eine richtige Vorhersage getroffen werden, während in nur 4% der Fälle eine falsche Entscheidung getroffen wurde.

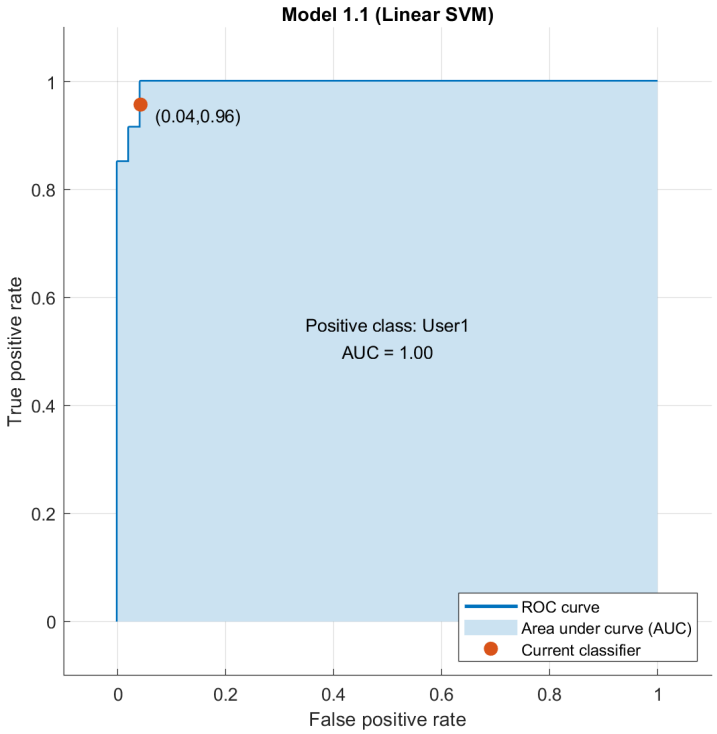

Abbildung 3.27: Plot der ROC zwischen User 1 und User 2

Die Confusions Matrix kann auch als Receiver operating characteristic (dt. Grenzwertoptimierungskurve) (ROC)-Plott dargestellt werden. Die Abbildung 3.27 stellt das ROC-Plott der gleichen Klassifizierung dar. Der rote Punkt zeigt die Genauigkeit des Klassifikators, in diesem Fall eine Genauigkeit von 96% und ist gleichzeitig die Fläche unter dieser Kurve. Ein perfektes Ergebnis würde einen rechteckigen Graphen mit einer Fläche von 1 (100%) darstellen [17, S.23-61].

Die gesamten Ergebnisse von Kandidat 1 und Kandidat 2 werden im nächsten Kapitel zusammen mit den Ergebnissen aller Kandidaten vorgestellt.

3.7.2 Gesamtübersicht der Ergebnisse

In diesem Abschnitt wird die Gasamtsauswertung präsentiert. Es wurde für alle Kandidaten aus Tabelle 3.1 eine Klassifizierung mit der im Abschnitt 3.5 präsentierten Klassifikatoren durchgeführt. Der SVM mit Sigmoid-Kernelfunktion wurde nicht untersucht, weil MATLAB eine solche Funktion nicht liefert.

Die Untersuchung erfolgt nach folgendem Schema:

Jeder Kandidat tritt zuerst als der echte User jeweils einzeln gegen alle anderen Kandidaten an. Der Kandidat tritt auch gegen sich selbst an. Das stellt ein sinnvolles Szenario dar, da dieser Fall im echten Verlauf am meisten auftreten wird. Es wurde jeweils die Genauigkeit und die Zeitdauer für jede Klassifikation aufgenommen.

Daraus sind Tabelle 3.3, 3.4, 3.5, 3.6 und 3.7 entstanden.

Kandidat	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
User 1	0	0.97872	0.95745	0.90426	0.96809	0.98936	0.94681	0.93617	0.92553	0.94681
User 2	0.98936	0	0.97872	0.96809	0.93617	0.97872	0.90426	0.97872	0.94681	0.98936
User 3	0.98936	0.97872	0	0.96809	0.97872	0.98936	0.95745	0.96809	0.96809	0.97872
User 4	0.90426	0.96809	0.96809	0	0.82979	0.96809	0.94681	0.91489	0.91489	0.97872
User 5	0.97872	0.95745	0.96809	0.81915	0	0.97872	0.93617	0.90426	0.94681	0.94681
User 6	0.98936	0.96809	0.98936	0.96809	0.96809	0	0.95745	0.94681	0.94681	0.91489
User 7	0.97872	0.91489	0.93617	0.93617	0.94681	0.95745	0	0.93617	0.88298	0.95745
User 8	0.92553	0.96809	0.97872	0.91489	0.88298	0.94681	0.96809	0	0.92553	0.92553
User 9	0.94681	0.94681	0.96809	0.92553	0.94681	0.94681	0.85106	0.90426	0	0.94681
User 10	0.96809	0.98936	0.97872	0.96809	0.94681	0.93617	0.95745	0.93617	0.95745	0

Tabelle 3.3: Ergebnis des Linearen SVM Klassifikators

Kandidat	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
User 1	0	0.95745	0.95745	0.90426	0.96809	0.96809	0.93617	0.95745	0.95745	0.94681
User 2	0.96809	0	0.95745	0.92553	0.91489	0.94681	0.90426	0.90426	0.92553	0.96809
User 3	0.97872	0.95745	0	0.96809	0.95745	0.97872	0.92553	0.95745	0.94681	0.96809
User 4	0.93617	0.95745	0.97872	0	0.88298	0.95745	0.94681	0.89362	0.90426	0.95745
User 5	0.96809	0.92553	0.96809	0.90426	0	0.96809	0.92553	0.91489	0.90426	0.96809
User 6	0.97872	0.96809	0.97872	0.95745	0.93617	0	0.96809	0.95745	0.95745	0.93617
User 7	0.93617	0.90426	0.92553	0.92553	0.90426	0.94681	0	0.92553	0.81915	0.94681
User 8	0.92553	0.96809	0.95745	0.89362	0.91489	0.95745	0.96809	0	0.88298	0.94681
User 9	0.92553	0.93617	0.94681	0.90426	0.91489	0.95745	0.80851	0.87234	0	0.94681
User 10	0.96809	0.96809	0.96809	0.95745	0.97872	0.94681	0.95745	0.96809	0.93617	0

Tabelle 3.4: Ergebnis des Gaussian SVM Klassifikators

Kandidat	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
User 1	0	0.96809	0.98936	0.95745	0.96809	0.98936	0.96809	0.92553	0.94681	0.95745
User 2	0.96809	0	0.95745	0.96809	0.96809	0.96809	0.90426	0.98936	0.94681	0.98936
User 3	0.98936	0.95745	0	0.97872	0.97872	0.98936	0.95745	0.97872	0.96809	0.97872
User 4	0.95745	0.95745	0.97872	0	0.90426	0.95745	0.92553	0.91489	0.91489	0.95745
User 5	0.95745	0.96809	0.97872	0.91489	0	0.94681	0.95745	0.91489	0.95745	0.96809
User 6	0.98936	0.95745	0.98936	0.95745	0.94681	0	0.93617	0.93617	0.92553	0.94681
User 7	0.96809	0.91489	0.96809	0.94681	0.94681	0.95745	0	0.94681	0.8617	0.96809
User 8	0.90426	0.97872	0.96809	0.89362	0.92553	0.92553	0.96809	0	0.90426	0.95745
User 9	0.94681	0.92553	0.95745	0.93617	0.95745	0.94681	0.84043	0.89362	0	0.91489
User 10	0.94681	0.98936	0.98936	0.96809	0.96809	0.95745	0.95745	0.95745	0.94681	0

Tabelle 3.5: Ergebnis des Polynomiale SVM Klassifikators

Kandidat	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
User 1	0	0.93617	0.95745	0.89362	0.89362	0.97872	0.92553	0.92553	0.89362	0.96809
User 2	0.93617	0	0.98936	0.93617	0.90426	0.96809	0.90426	0.8617	0.8617	0.96809
User 3	0.94681	0.95745	0	0.97872	0.96809	0.96809	0.97872	0.94681	0.95745	0.96809
User 4	0.89362	0.95745	0.97872	0	0.81915	0.95745	0.96809	0.88298	0.89362	0.93617
User 5	0.90426	0.89362	0.95745	0.81915	0	0.95745	0.90426	0.8617	0.91489	0.96809
User 6	0.97872	0.96809	0.97872	0.95745	0.93617	0	0.94681	0.96809	0.92553	0.8617
User 7	0.92553	0.8617	0.94681	0.96809	0.92553	0.95745	0	0.96809	0.79787	0.94681
User 8	0.90426	0.89362	0.95745	0.90426	0.85106	0.97872	0.94681	0	0.8617	0.94681
User 9	0.90426	0.8617	0.95745	0.88298	0.90426	0.93617	0.76596	0.88298	0	0.90426
User 10	0.97872	0.96809	0.95745	0.92553	0.94681	0.8617	0.93617	0.95745	0.90426	0

Tabelle 3.6: Ergebnis des k-NN Klassifikators

Die Diagonale der Tabelle stellt die Ergebnisse der auftretenden Kandidaten gegen sich selbst dar. Eine null bedeutet, dass keine Unterscheidung der Daten stattfinden konnte. Das Ergebnis war auch zu erwarten.

Die Abbildung 3.28 stellt die Tabellendaten graphisch dar.

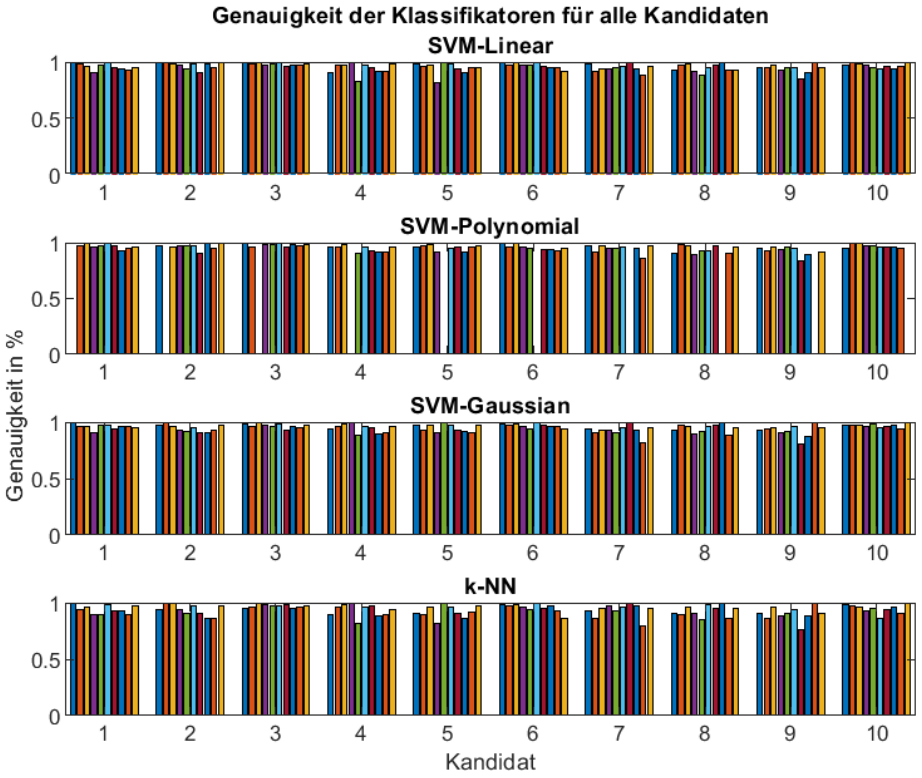

Abbildung 3.28: Plot der Ergebnisse aus Tabelle 3.3, 3.4, 3.5, und 3.6

,

Die Genauigkeit liegt im Durchschnitt deutlich über 90%. Die schlechtesten Ergebnisse liegen bei ca. 81% und werden als Ausnahme betrachtet.

Nachfolgend wird die notwendige Zeit für die Lern- und Testphase der Klassifikatoren präsentiert.

Da die Ausführungszeit ungefähr gleich für alle Kandidaten ausgefallen ist und es zudem unnötig wäre noch eine lange Tabelle mit Zeitergebnissen einzufügen, werden für jeden Klassifikatoren nur die Eckzeiten angegeben.

Die langsamsten und schnellsten gemessenen Zeiten sowie eine Durchschnittszeit über alle Testversuche pro Klassifikator werden in der Tabelle 3.7 präsentiert.

	Schnellste Zeit	**Langsamste Zeit**	**Durchschnittliche Zeit**
Linearen SVM Klassifizierer	0.013708 s	0.12673 s	0.021459 s
Gaussian SVM Klassifikator	0.013761 s	0.065801 s	0.021195 s
Polynomiale SVM Klassifikator	0.013804 s	0.097905 s	0.017898 s
k-NN Klassifikator	0.011098 s	0.21223 s	0.018973 s

Tabelle 3.7: Ergebnis des Klassifikatorenausführungszeiten

Die Zeiten aus Tabelle 3.7 entsprechen nur der Zeit, welche notwendig für den Modellaufbau und die Kreuzkorrelation ist. Die Zeit für die Erstellung des Templates wurde nicht gemessen.

Parameterauswahl für die Implementierung

Anhand der Ergebnisse aus den Abschnitten 3.7.1 und 3.7.2 wurde für die Implementierung ein SVM-Klassifikator ausgewählt. Es werden für die Implementierung als auch der Modellierung die gleichen Merkmale verwendet. Was die Kernelfunktionen und Kernelparameter betrifft, wird für die Implementierung eine Linearen-Kernelfunktion untersucht.

4 Implementierung auf einem Android Smartphone

In diesem Kapitel erfolgt die Erstellung der ANDROID-APK Software unter Verwendung der JAVA-Programmiersprache. Die tatsächliche Implementierung der Klassifizierung erfolgt unter dem gleichen Schema wie im Kapitel 3 vorgestellt. Ein Funktionstest und eine Auswertung der Applikation wird im Abschnitt 4.4 präsentiert.

4.1 Übersicht

4.1.1 Allgemein

Für die Erstellung dieser Software wurden mehrere externe Bibliotheken und Funktionen verwendet, die an dieser Stelle erwähnt werden. Für diverse mathematische Operationen wurde die Bibliothek `commons-math3-3.6.1.jar` verwendet [6]. Diese Funktionen wurden grundsätzlich bei der Extraktion der Merkmale verwendet. Für die Fourier Transformation wurde der Algorithmus `fft.c` von Douglas L. Jones angepasst [8]. Anschließend wurde für die Klassifizierung und für die Evaluierung von biometrischen Merkmalen die `weka-stable-3.8.1-SNAPSHOT.jar` Bibliothek aus der WEKA-Software verwendet [11] und für den linearen SVM wurde die `libsvm-1.0.3.jar` Bibliothek benutzt [12] [5].

4.1.2 Programmablauf

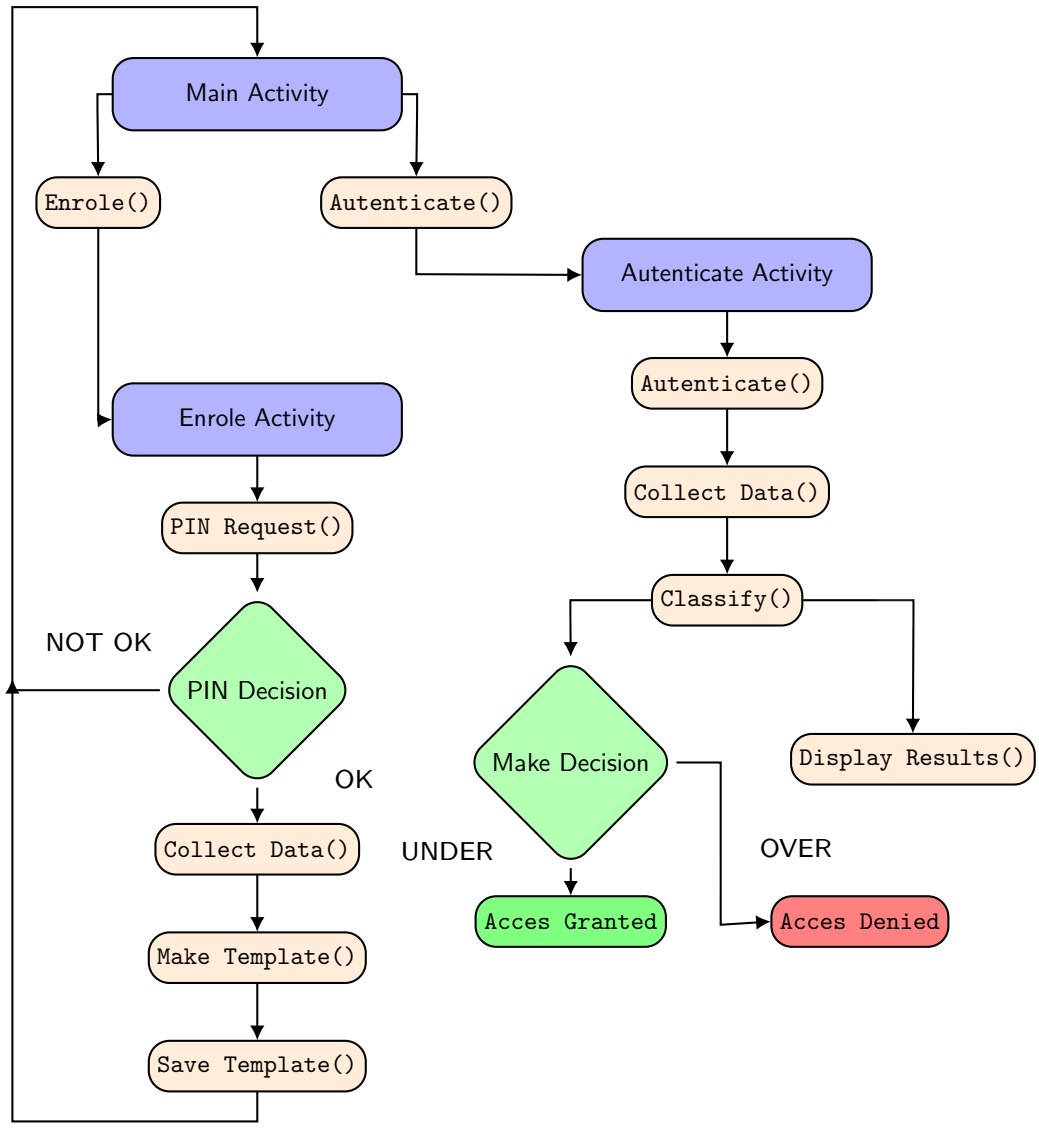

Abbildung 4.1: Programmablauf

Die Abbildung 4.1 stellt den JAVA-Applikationsablauf graphisch dar. Eine Erläuterung des Programmablaufs erfolgt im Abschnitt 4.2 anhand der GUI-Oberfläche.

4.2 GUI der JAVA-Applikation

Als Nächstes wird der Programmablauf anhand der GUI-Oberfläche präsentiert. Beim Starten des Programmes wird zuerst die `Main Activity` erstellt. Ab hier gibt es die Möglichkeit eine Authentifizierung durchzuführen oder einen neuen User zu speichern wie in der Abbildung 4.2 dargestellt.

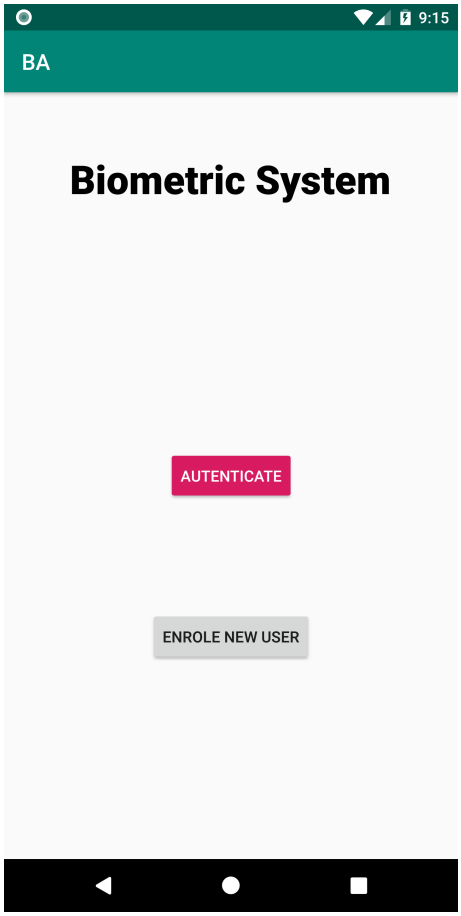

Abbildung 4.2: Applikation: Autentication Ansicht

Wenn danach ein neuer User gespeichert werden soll, wird die Betätigung der `(ENROLE NEW USER)` dahinführen. Um diese Aktion durchführen zu können wird der Anwender nach einer PIN gefragt. Die Abbildung 4.3 veranschaulicht diesen Prozess. Für Testzwecke wurde die PIN: `0000` ausgewählt.

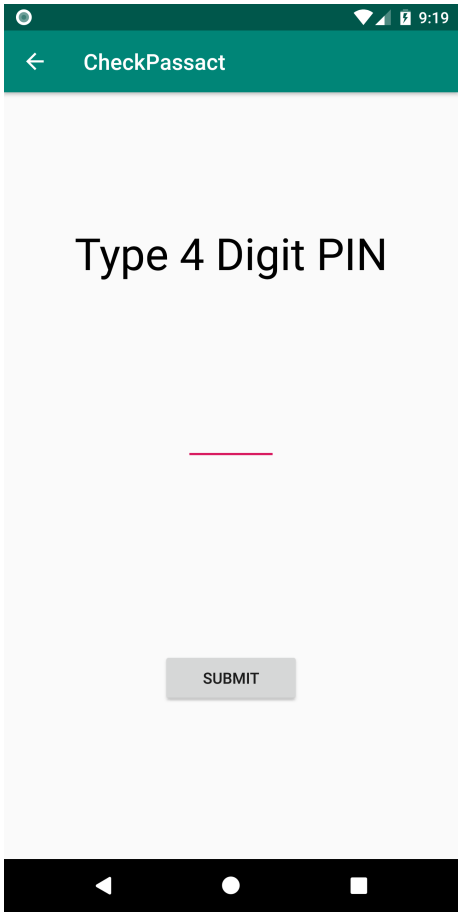

Abbildung 4.3: Applikation: PIN Ansicht

Bei einer erfolgreichen Angabe der PIN wird der Anwender zur `(Enrole Activity)` hingeführt. Diese GUI-Oberfläche ist in der Abbildung 4.4 dargestellt. Bei einer nicht erfolgreichen Angabe erfolgt nur eine Meldung, dass die PIN-Angabe fehlerhaft war. Mit dem Pfeil links oben in der Abbildung 4.3 wird der Anwender zu der `Main Activity` hingeführt.

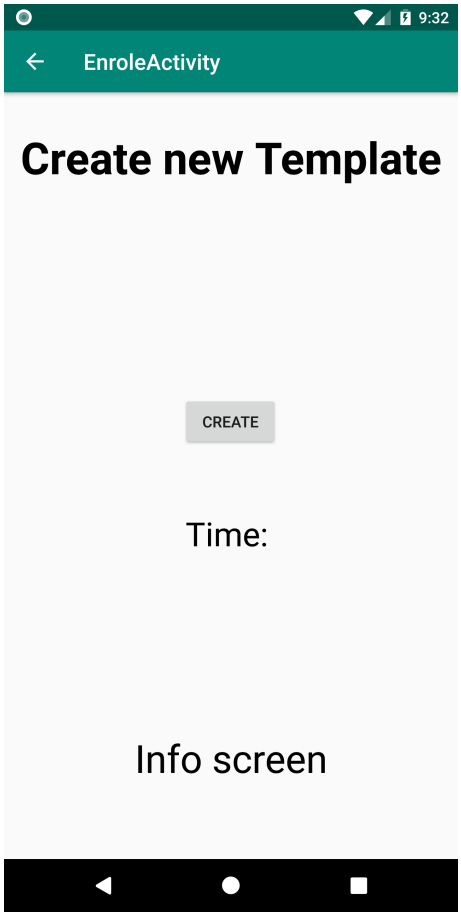

Abbildung 4.4: Applikation: Template erstellen Ansicht

In der `Enrole Activity` gibt es die Möglichkeit über den Pfeil links oben zurück zu der `Main Activity` zu gelangen. Die andere Möglichkeit wäre, eine neuen Template zu erstellen bzw. einen neuen User als Hauptuser zu speichern. Dieser Prozess wird durch Betätigung der `(CREATE)` Knopf fortgeführt. Der Anwender wird zunächst hingewiesen das Laufen anzufangen. Nach 30 Sekunden wird die Laufaufzeichnung beendet. Mit Hilfe der aufgezeichneten biometrischen Daten wird im Hintergrund die Template mit Hilfe der Klassen `Make Template()` und `Save Template()` erstellt und gespeichert. Der Anwender wird bei der Beendigung dieser Prozesse mit Hilfe einer Anzeige in den `Info screen` über das positive oder negative Ergebnis dieses Prozesses benachrichtigt.

Als Nächstes wird die `Autenticate Act` vorgestellt. Eine Ansicht der GUI-Oberfläche ist in der Abbildung 4.5 dargestellt. Hier gelangt der Anwender nur aus der `Main`

`Activity`-Ansicht, indem der `(AUTENTICATE)` Knopf gedrückt wird. Aus dieser Ansicht gibt es die Möglichkeit entweder zurück zur `Main Activity` über den Pfeil links oben zu navigieren oder mit Betätigung des `(Autenticate)` Knopfes eine Authentifizierung durchzuführen.

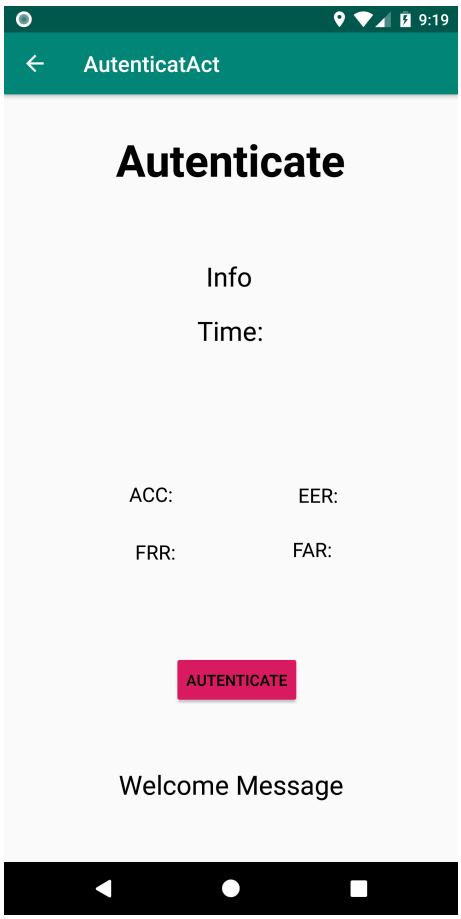

Abbildung 4.5: Applikation: Autentication Ansicht

Der Authentifizierungsprozess wird wie folgt durchgeführt:

Zuerst wird der Anwender über eine Anzeige benachrichtigt das Laufen anzufangen. Die Zeit wird in dem Feld `Time:` angezeigt. Nach 30 Sekunden wird die Aufzeichnung beendet und der Klassifizierungsprozess `classify()` wird aufgerufen.

Bei Beendigung dieses Prozesses werden in den vorgesehenen Feldern die ACC, EER, FRR, FAR und die notwendige Zeit für die Klassifizierung angezeigt. Parallel dazu erfolgt die Auswertung der Klassifizierung. Es wurde eine Grenze (Threshold) von 85% gesetzt.

Alle Ergebnisse, welche diese Grenze überschreiten werden als nicht der richtiger User bewertet. Alle Ergebnisse die unter diese Grenze fallen werden als richtige User interpretiert. Das Ergebnis wird als Nachricht auf der GUI-Oberfläche angezeigt. Die zwei möglichen Ergebnisse sind in Abbildung 4.6 dargestellt. Links ist eine positive Bewertung angezeigt, während rechts die negative Authentifizierung angezeigt wird.

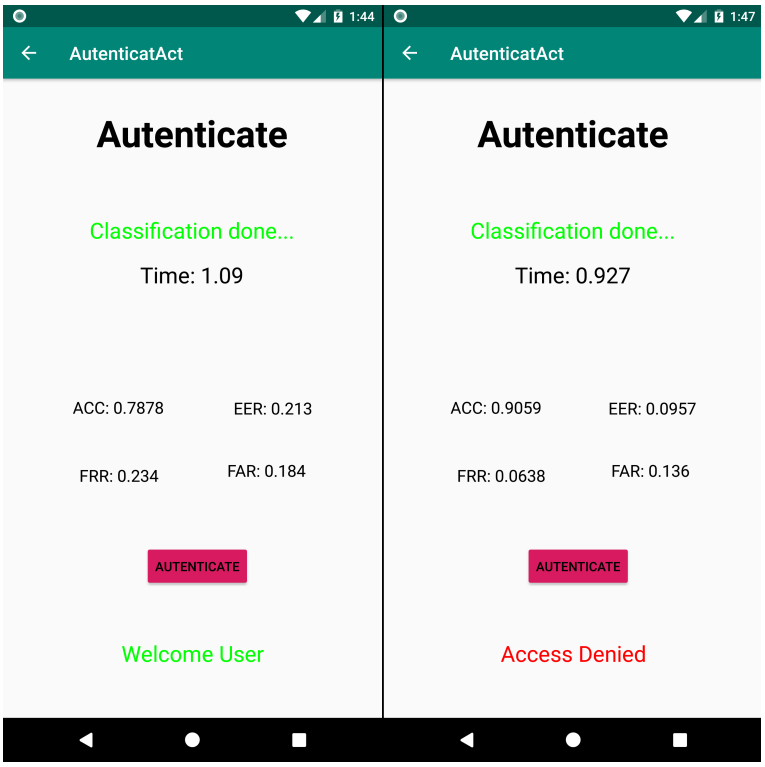

Abbildung 4.6: Applikation: Ergebnis der Authentifizierung

4.3 Optimierung

Die ersten ANDROID-Softwaretestversuche erzielten für die Klassifikation keine guten Ergebnisse. Das lag daran, dass das LibSVM-Modell nicht mit den richtigen Parametern initialisiert wurde. Bei der Erstellung des LibSVM-Modells werden ohne weitere Angaben die Standardparameter ausgewählt. Diese Parameter haben nicht mit den Werten im Abschnitt 3.7.2 übereingestimmt.

Zusätzlich wurde eine Optimierung der Kernel-Parameter vorgenommen. Diese wurden konform der Anweisungen in der WEKA-Hilfestellung für LibSVM [23] durchgeführt. Für

die Optimierung wurde ein sogenanntes (gridSearch) durchgeführt. Daraus sind folgende Kernelparameter entstanden.

- setGamma(0.01);
- setCost(1000);
- setEps(0.001);

Anhand dieser Änderungen war es möglich ähnliche Ergebnisse wie im Kapitel 3 zu erzielen. Um die Programmzeiten zu optimieren, werden in Java für die Berechnung der Eigenschaften, für die Erstellung von Templates und für die Klassifizierung jeweils parallele Threads (`AsyncTask`) verwendet. Diese ermöglichen die gleichzeitige Verarbeitung von mehreren Prozessen und erschaffen somit eine Zeitersparnis.

4.4 Funktionstest und Auswertung

In diesem Kapitel werden die Ergebnisse der Funktionstest und eine Auswertung präsentiert.

Ein Funktionstest der GUI-Oberfläche wurde durchgeführt. Der Ablauf aus der Abbildung 4.1 konnte problemlos bestätigt werden. Die GUI-Anzeige und GUI-Knopf Bedienung funktioniert auch regelkonform. Die Ergebnisse der GUI-Anzeige können mit Hilfe der Abbildungen 4.4, 4.5 und 4.6 aus Abschnitt 4.2 bestätigt werden.

Für die Auswertung der Klassifizierung wurde eine Authentifizierung mit der aus dem Abschnitt 1.2 vorgestellten Rahmenbedingungen für alle 10 Kandidaten durchgeführt. Es treten für diesen Testversuch alle Kandidaten einmal gegen den User 1 an. Es wurden die Zeiten, die Genauigkeit und die Fehlerraten aus Abschnitt 2.3.4 gemessen.

Die Ergebnisse dieser Testversuche sind in der Tabelle 4.1 dargestellt.

Kandidat	ACC|User 1	Zeit	EER	FAR	FRR
User 1	1.0638	1.063 s	0.0	0.0	0.0
User 2	100	0.989 s	0.0	0.0	0.0
User 3	96.83	1.043 s	0.031	0.020	0.042
User 4	98.935	1.087 s	0.010	0.0	0.021
User 5	98.936	1.101 s	0.010	0.0	0.021
User 6	95.744	1.084 s	0.042	0.042	0.042
User 7	91.489	0.992 s	0.085	0.111	0.063
User 8	90.425	1.247 s	0.095	0.136	0.063
User 9	90.425	0.958 s	0.095	0.108	0.085
User 10	97.872	1.025 s	0.021	0.021	0.021

Tabelle 4.1: Ergebnisse der Applikation

Die Ergebnisse aus der Tabelle 4.1 entsprechen bei der Einhaltung der Rahmenbedingungen die vorgestellten Erwartungen. Die Ausführungszeiten sind im Bereich von einer Sekunde, welches einem guten Ergebnis entspricht. Die Genauigkeit bei der Untersuchung der Kandidaten ist für alle 10 Fälle mit Ausnahme von User 1 über 90% ausgefallen. Das Ergebnis von 1.0683% Genauigkeit bestätigt, dass es sich bei der Untersuchung um den gleichen User handelt. Die EER und die anderen Fehlerraten bestätigen auch die erwarteten Ergebnisse.

Mit den Ergebnissen aus Tabelle 4.1 konnten die theoretischen MATLAB-Modell-Ergebnisse aus dem Kapitel 3 bestätigt werden.

5 Abschluss der Arbeit

Das folgende und letzte Kapitel fasst den Inhalt dieser Arbeit und die Erkenntnisse kurz zusammen. Es wird auf die Zielsetzung aus Abschnitt 1.2 und auf die Umsetzung der gestellten Anforderungen eingegangen.

Es erfolgt außerdem ein Ausblick für eine Erweiterung und Verbesserung des implementierten biometrischen Authentifizierungssystems.

5.1 Fazit

In dieser Arbeit wurde ein Konzept für ein biometrisches Authentifizierungssystem mit Hilfe der Ganganalyse in Form einer Smartphone ANDROID-Applikation realisiert. Um diese Software realisieren zu können, wurde die Arbeit in aufeinander bauende Abschnitte unterteilt. Zuerst wurden durch Literaturrecherche die notwendigen Grundlagen für diese Arbeit Schritt für Schritt erarbeitet.

Danach wurde ein Konzept rausgesucht mit dem experimentiert wurde. Leider hat sich die erste ausgewählte Einsatzmethode, die auf Kohärenz basiert, als nicht geeignet für diese Arbeit gezeigt. Nach zwei Wochen Untersuchung mit enttäuschenden Ergebnissen wurde somit diese Methode aufgegeben.

Nach weiteren Recherchen wurden auch andere Methoden untersucht. Durch erfolgreiche Testversuche wurde das maschinelle Lernen als Hauptmethode für das biometrische Authentifizierungsverfahren ausgewählt. Mit einer vielversprechenden Idee, belegt durch erfolgreiche Testversuche, wurden daraufhin die konkreten Ziele festgelegt.

Der nächste Schritt war das Erstellen eines Testmodells mit Hilfe der Software MATLAB. Hier wurde mit Hilfe von echten biometrischen Daten, aufgenommen von zehn Kandidaten, experimentiert. Verschiedene Datenverarbeitungsmethoden und Klassifikatoren

wurden anhand der Testdaten untersucht. Da das nächste Kapitel stark von den Ergebnissen dieses experimentellen Modells abhing, nahm es sehr viel Zeit bei der Auswahl der Funktionen, Klassifikatoren und der Auswertung der Ergebnisse in Anspruch.

Die erfolgreiche Realisierung dieses Konzeptes ist auch ein großer Teil der Zielsetzung dieser Arbeit. Die festgelegten Ziele für den Modellaufbau aus Abschnitt 1.2 konnten sehr gut eingehalten werden. Die Ergebnisse für das MATLAB-Modell aus Abschnitt 3.7 verdeutlichen erneut die Einhaltung der Zielsetzung bei der Erstellung des MATLAB-Modells. Somit wurden die Ziele sowie die Datenauslesung und die richtige Auswertung in den vorgegebenen Parametern eingehalten.

Mit einer durchschnittlichen Genauigkeit von deutlich über 90%, welches positiv überraschte, und einer Klassifizierungszeit von unter eine Sekunde, konnten auch diese Ziele eingehalten werden.

Danach blieb nur die Implementierung als ANDROID-Applikation übrig. Diese wurde mit Hilfe der ausgewählten Parameter aus Abschnitt 3.7.2 durchgeführt.

Da für die Implementierung nicht mehr viel Zeit übrig blieb, wurde hierbei nur das Notwendige für die Erreichung der Zielsetzung betrachtet. Der Funktionstest und die Ergebnisse aus Abschnitt 4.4 bestätigen die im Abschnitt 1.2 gesetzten Ziele für die Implementierung. Es konnten somit bei der Implementierung das Ziel von Einhaltung der Ergebnisse wie bei den theoretischen MATLAB-Modell eingehalten werden. Auch das Ziel mit der Anzeige und Steuerung der Applikation konnte in der GUI-Oberfläche problemlos eingebaut werden. Die Kernaussage dieser Thesis ist, dass eine biometrische Authentifizierung mittels Ganganalyse auf ein Smartphone möglich ist. Das ist jedoch ein sehr komplexer Prozess und für eine Echtzeiteinsetzung auf einem Smartphone müssen viele andere Szenarien betrachtet werden.

5.2 Ausblick

Bei der Erstellung der Arbeit wurden die Raumbedingungen des biometrischen Authentifizierungssystems mittels Ganganalyse deutlich eingegrenzt. Die Rahmenbedingungen, auf denen diese Arbeit basiert, entsprechen nur einem Teil eines reellen einsetzbaren Systems. Folgende Verbesserungen oder Funktionserweiterungen können im Rahmen einer zukünftigen Arbeit betrachtet werden:

- Eine Aktivitätserkennung könnte implementiert werden, womit die Einsatzmöglichkeiten deutlich erweitert werden würden
- Für eine Erhöhung der Genauigkeit könnten mehrere Templates für den echten User erstellt werden

Die ANDROID-Applikation könnte um einiges erweitert werden mit Funktionen zum Beispiel:

- Datenverschlüsselung
- Multi-User Erweiterung
- extra Schutz mit einem Passwort oder PIN
- Authentifizierung im Hintergrund bei bestimmten getriggerten Prozessen

Literaturverzeichnis

[1] *Android Studio*. https://developer.android.com/studio/intro/. – Eingesehen am 07.04.2019.

[2] *Mathworks Matlab Geschichte*. https://de.mathworks.com/company.html?s_tid=hp_ff_a_company. – Eingesehen am 25.03.2019.

[3] Aminian, Kamiar ; Najafi, Bita ; Büla, Christophe ; Leyvraz, P.-F ; Robert, Ph: Ambulatory Gait Analysis Using Gyroscopes. In: *25th Annual Meeting of the American Society of Biomechanics* (2001), 01.

[4] Bishop, Christopher M.: *Pattern Recognition and Maschine Learning*. Springer, 2006. – ISBN 0-387-31073-8.

[5] Chang, Chih-Chung ; Lin, Chih-Jen: *LIBSVM - A Library for Support Vector Machines*. 2001. – URL http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[6] Commons, Appache: *Commons Math: The Apache Commons Mathematics Library*. – URL https://commons.apache.org/proper/commons-math/.

[7] David G. Stork, Richard O. D. und Peter E. Hart und: *Pattern Classification*. Springer, 2006. – ISBN 0-471-05669-3.

[8] Douglas L. Jones, University of I.: *FFT: in-place radix-2 DIT DFT of a complex input*. 02 1992. – URL http://cnx.rice.edu/content/m12016/latest.

[9] Däs, Sebastian: *Compliance-konforme Einbindung biometrischer Authentifizierungssysteme in das betriebliche IT-Sicherheitsmanagement*. Wiesbaden: Springer Gabler, 2014. – URL https://doi.org/10.1007/978-3-658-23466-9. – ISBN 978-3-658-23465-2.

[10] Eckert, Claudia: *IT-Sicherheit, 9.Auflage*. Oldenburg: De Gruyter, 2014. – ISBN 978-3-486-77848-9.

[11] Eibe Frank, und Ian H. W.: *WEKA-Software*. 2016.

[12] EL-Manzalawy, Yasser: *WLSVM*. 2005. – URL http://www.cs.iastate.edu/~yasser/wlsvm/.

[13] Götz-Neumann, Kirsten: *Gehen verstehen: Ganganalyse in der Physiotherapie*. Georg Thieme Verlag, 2013. – URL http://www.gehen-verstehen.net/. – ISBN 13: 3-13-132371-2.

[14] Hewlett-Packard Company, und a.: Universal Serial Bus 3.0 Specification. (2008), 09.. – URL http://www.usb.org.

[15] Konstantinos Kountroumbas, Sergios T. und: *Pattern Recognition*. Academic Press, 1999. – ISBN 0-12-686140-4.

[16] Marc Kraft, Catherine Disselhorst-Klug und a.: *Rehabilitationstechnik*. De Gruyter, August 2015. – URL https://doi.org/10.1515/9783110252262. – ISBN 978-3-11-025226-2.

[17] Mathworks: *Statistics and Machine Learning Toolbox Users Guide*.

[18] Osterhage, Wolfgang W.: *Sicherheitskonzepte in der mobilen Kommunikation*. Wiesbaden: Springer Gabler, 2018.. – URL https://doi.org/10.1007/978-3-662-57903-9. – ISBN 978-3-662-57902-2.

[19] RRZN, Dr. Thomas Kröckertskothen /.: *Java 2 Grundlagen und Einführung, 6.Auflage*. RRZN Hanover, 2007. – URL www.rrzn.uni-hannover.de/buecher.html. – ISBN 4032141001079.

[20] STMicroelectronics: *LSM6DSL Datenblatt*. – URL www.st.com.

[21] Thingom Bishal, NathA. V. N.: Person Recognition using Smartphones' Accelerometer Data. In: *25th Annual Meeting of the American Society of Biomechanics* (2017), 11.

[22] Trojahn, Matthias: *Sichere Multi-Faktor-Authentifizierung an Smartphones mithilfe des Tipverhaltens*. Wiesbaden: Springer Gabler, 2016. – URL http://dnb.d-nb.de. – ISBN 978-3-658-14048-9.

[23] Witten, Ian: Advanced Data Mining with Weka (Class 3 – Lesson 1). In: *PDF*

[24] Witten, Ian H. ; Frank, Eibe: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005. – URL http://www.cs.waikato.ac.nz/~ml/weka/book.html.

[25] *Studie Zenith Mobile Advertising*. – URL https://www.zenithmedia.com/smartphone-penetration-reach-66-2018. – Eingesehen am 07.03.2019.

6 Beigefügte CD

Die beigefügte CD kann bei Erstprüfer Prof. Dr.-Ing. Robert Fitz oder Zweitprüfer Prof. Dr. -Ing. Mark Hensel eingesehen werden. Auf der CD befinden sich folgende Dateien:

- Ordner Bachelorthesis
  - Bachelorthesis als PDF-Datei
- Ordner Android
  - Android Studio Projekt inkl. verwendete Bibliotheken
- Ordner Csv_Dataset
  - biometrisch Daten aller 10 Kandidaten als .csv
- Ordner Matlab
  - Matlab Projekt inkl. Matlab Bilder

Glossar

Android Studio ist ein JAVA Editor.

Android ist sowohl ein Betriebssystem als auch eine Software-Plattform für Smartphones. Basis ist der Linux-Kernel.

Apple ist ein US-Amerikanisches Computer Unternehmen, die unter anderen sich mit Smartphones beschäftigt. Der Kern von Apple iOS ist ein angepasstes macOS, welches das Betriebssystem des Macs ist.

Java ist eine objektorientierte Programmiersprache und eine eingetragene Marke des Unternehmens Sun Microsystems..

Matlab ist eine kommerzielle Software des US-amerikanischen Unternehmens MathWorks zur Lösung mathematischer Probleme und zur grafischen Darstellung der Ergebnisse..

Bluetooth beschreibt eine Datenübertragungsprotokoll zwischen Geräten über kurze Distanz per Funktechnik.

IntelliJ ist ein integrierte Entwicklungsumgebung für Entwicklung von Computer Software.

STMicroelectronics ist ein globales Halbleiterunternehmen.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Ort	Datum	Unterschrift im Original