Hochschule Darmstadt

Fachbereiche Mathematik und
Naturwissenschaften & Informatik

Spark-basierte Analyse von Monitoring-Daten intensiv überwachter Patienten zur Identifikation von Anomalien: Ein Use Case der Charité Health Data Platform

Abschlussarbeit zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

im Studiengang Data Science

vorgelegt von

Linda Rebstadt

Matrikelnummer: 760367

Referentin	:	Prof. Dr. Antje Jahn
Korreferent	:	Prof. Dr. Arnim Malcherek

Ausgabedatum	:	01.09.2020
Abgabedatum	:	12.02.2021

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 12. Februar 2021

Linda Rebstadt

ZUSAMMENFASSUNG

MOTIVATION: Herz-Kreislauferkrankungen sind die häufigste Todesursache der Welt, eine frühzeitige Erkennung kann dem entgegenwirken. Zur Erforschung solcher kardiovaskulärer Anomalien spielt die Analyse von Elektrokardiogrammen (EKG)s eine entscheidende Rolle. Durch die Anschaffung des Data Warehouse Connects und der Nutzung der Health Data Plattform ist es der Neurologie der Charité möglich, EKGs in hoher Auflösung über lange Zeiträume für die Forschung vorzuhalten. Jedoch stellen die großen Datenmengen und die technischen Artefakte Herausforderungen bei der Analyse dar. Ziele sind daher das Finden einer Methode zur Elimination der technischen Artefakte *Baseline Wander* und *Power-Line-Interferenz* sowie das Entwickeln eines Modells zur Erkennung abnormaler Herzschläge, um die Datenmenge auf die relevanten Stellen zu begrenzen.

METHODEN: Zur Elimination der technischen Artefakte wurden Dekompositionen, Filter und Transformationen evaluiert, die auf diese großen Datenmengen mit moderatem Rechenaufwand zuverlässig arbeiten. Das Modell zur Herzschlagerkennung setzt zur Reduktion des Aufwands auf den neuen Daten, auf die Prinzipien des *Transfer Learnings*. Es wurden Machine-Learning-Modelle auf der MIT-BIH-Arrhythmia-Datenbank trainiert, anschließend auf die Charité-Daten übertragen und anhand einer Teilmenge der vorhandenen Monitoring-Annotationen evaluiert.

ERGEBNISSE: Die Methoden zur Elimination technischer Artefakte wurden auf Basis von vier Metriken verglichen: Signal-to-noise ratio (SNR), Maximum Absolut Error (MAX), Normalised Cross Correlation (NCC) und Mean Squared Error (MSE). Das beste Ergebnis erreichte die Sparse Signal Decomposition (SSD) mit einem SNR von 12,43, einem MAX von 0,15, einer NCC von 0,96 und einem MSE von 0,003. Das nächstbeste Ergebnis erreichte die Diskrete Wavelet-Transfor-mation (DWT) mit einem dmey-Wavelet als Basisfunktion. Bei ihr lag der SNR bei 10,34, der MAX bei 0,31, der NCC bei 0,94 und der MSE bei 0.004. Den geringsten Rechenaufwand zeigte mit Abstand die DWT. Das Modell zur Erkennung abnormaler Herzschläge zeigte eine Accuracy von 99,13 % auf der MIT-BIH-Arrhythmia-Datenbank. Auf den Charité-Daten konnte jedoch nur eine Accuracy von 57,86 % erreicht werden.

FAZIT: Die besten Ergebnisse zur Elimination der technischen Artefakte zeigte die SSD. Die DWT sollte dennoch in Betracht gezogen werden, da der Rechenaufwand deutlich geringer ist. Die Herzschlagerkennung weiste auf dem MIT-BIH-Datensatz eine hohe Accuracy auf. Die Übertragung des Modells auf die Charité-Testdaten zeigte noch keine zuverlässigen Ergebnisse.

ABSTRACT

PURPOSE: Cardiovascular diseases (CVDs) are the leading cause of death world wide. Early detection could significantly reduce the mortality caused by CVDs. ECG analysis plays a critical role in the study of cardiovascular abnormalities. By acquiring the Data Warehouse Connect and using the Health Data Platform, Neurology of Charité is able to maintain ECGs at high resolution over long periods of time for research. However, the large data volumes and technical artifacts pose challenges in analysis. Therefore, the goals are to find a method to eliminate the technical artifacts of *Baseline Wander* and *Power Line Interference*, and to develop a model for abnormal heartbeat detection to limit the amount of data to the relevant locations.

METHODS: To eliminate these technical artifacts, decompositions, filters, and transformations were evaluated that work reliably on these large data sets with moderate computational effort. The heartbeat detection model relies on the principles of *Transfer Learning* to minimize the overhead on new data. Machine-learning models were trained on the MIT-BIH Arrhythmia Database, applied to the Charité data and evaluated using a subset of the existing monitoring annotations.

RESULTS: Elimination methods were compared on the basis of four metrics: Signal-to-Noise Ratio (SNR), Maximum Absolute Error (MAX), Normalized Cross Correlation (NCC), and Mean Squared Error (MSE). The best result was achieved by Sparse Signal Decomposition (SSD) with an SNR of 12.43, a MAX of 0.15, an NCC of 0.96, and an MSE of 0.003. The next best result was achieved by the Discrete Wavelet Transform (DWT) with a dmey wavelet as the basis function. The SNR was 10.34, the MAX 0.31, the NCC 0.94, and the MSE 0.004. The model for detection of abnormal heartbeats showed an accuracy of 99.13 % on the MIT-BIH Arrhythmia database, but on the Charité data only an accuracy of 57.86 % could be achieved.

CONCLUSION: The SSD showed the best results for eliminating the technical artifacts. The DWT should still be considered, since the computational effort is significantly lower. The heartbeat detection shows a high accuracy on the MIT-BIH dataset. However, the transfer of the model to the Charité test data did not show reliable results.

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

LISTINGS

ABKÜRZUNGSVERZEICHNIS

AWGN Additive White Gaussian Noise

BW Baseline Wander

CEEMDAN Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

CWT Kontinuierliche Wavelet-Transformation

DFT Diskrete Fourier-Transformation

DKT Diskrete Kosinus-Transformation

DST Diskrete Sinus-Transformation

DWC Data Warehouse Connect

DWT Diskrete Wavelet-Transformation

DTCWT Dual Tree Complex Wavelet Transform

DTW Dynamic Time Warping

EEMD Ensemble Empirical Mode Decomposition

EKG Elektrokardiogramm

EMD Empirical Mode Decomposition

HDP Health Data Platform

ICA Independet Components Analysis

IMF Intrinsic Mode Functions

KNN K-Nächste-Nachbarn

LMS Least Mean Square

LDA Lineare Diskriminanzanalyse

NCC Normalised Cross Correlation

NLMS Normalized Least Mean Square

MA Muscle Artefact

MAX Maximum Absolute Error

MSE Mean Squared Error

PCA Hauptkomponentenanalyse

PLI Power-Line-Interferenz

PSD Power Spectral Density

QDF Quadratische Diskriminanzfunktion

RLS Recursive Least Square

SNR Signal-to-Noise Ratio

SVM Support Vector Machine

Teil I

THESIS

1 EINLEITUNG

Über 35% der Todesursachen 2019 sind in Deutschland auf Herz-Kreislauferkrankungen zurückzuführen laut Statistischem Bundesamt [9]. Auch im Jahr 2020 meldete das Robert-Koch-Institut sowohl bei Frauen als auch bei Männern Herz-Kreislauferkrankungen neben Krebs und Krankheiten des Atmungssystems als eine der häufigsten Todesursachen [58]. Neue Erkenntnisse zu Herzerkrankungen zu schaffen und medizinische Ableitungen aus den Erkenntnissen zu ziehen, würde aus diesem Grund einem Großteil der Gesellschaft zu Gute kommen. Die frühzeitige Erkennung von Krankheitsbildern oder das Aufdecken der Zusammenhänge zwischen den verschiedenen medizinischen Parametern lässt sich insbesondere in der Zeit der Digitalisierung und der computerbasierten Unterstützung schnell vorantreiben.

1.1 MOTIVATION

Das Potential der digitalen Auswertung hat die Neurologie der Charité erkannt und eine neue Plattform, Philips Data Warehouse Connect (DWC), zur Langzeitspeicherung der Monitoring-Daten der Intensivstationen und der Schlaganfallstationen etabliert. Die erhobenen Monitoring-Daten werden im DWC über lange Zeiträume gespeichert, können jedoch dort nur in sehr begrenztem Umfang analysiert und mit anderen Daten verknüpft werden. Um diese Restriktion zu überwinden und Langzeitfragestellungen beantworten zu können, werden die Daten auf die Health Data Platform der Charité übertragen. Dort können sie sowohl mit weiteren Datenquellen verknüpft als auch umfangreichen Big-Data-Analysen unterzogen werden. Diese Big-Data-Analysen unterstützen das Vorantreiben der medizinischen Forschung auf dem digitalen Weg.

Das Monitoring-System stellt über 90 aufgezeichnete Parameter bereit, die unterschiedlich oft von den Ärzten zur Überwachung der Patienten genutzt werden. Vier Parameter wurden zunächst von der Charité zur Untersuchung priorisiert: Das Elektrokardiogramm (EKG), der Blutdruck (arterieller Blutdruck und nicht-invasiver Blutdruck), die Sauerstoffsättigung und die Atemfrequenz. Im Umfang der Masterarbeit kann lediglich ein Parameter untersucht werden. Daher befasst sich die weitere Arbeit mit dem höchst priorisierten Parameter - Elektrokardiogramm -, welcher bei fast jedem Patienten aufgezeichnet wird.

Das EKG ist heutzutage das Standardwerkzeug zur Erkennung verschiedener kardiovaskuläre Anomalien und beschreibt eine nicht-invasive Untersuchungsmethode zur Erfassung der Herzaktion und Lage des Herzens, indem Spannungen an Elektroden gemessen werden [37, 77]. Langzeitanalysen auf dem EKG könnten daher hilfreiche Erkennisse zu Herzkrankheiten erzielen.

Auf den Intensivstationen und Schlaganfallstationen liegen Patienten, die teilweise über mehrere Wochen aufgrund verschiedenster Krankheiten überwacht werden, wodurch eine genaue Untersuchung der Entwicklung des EKGs auf Langzeitsicht möglich ist.

Ein so großer Datensatz, der stetig wächst, ist für die medizinischen Forscher*innen direkt nicht nützlich, da es sehr viel Zeit kostet lange EKG-Aufzeichnungen (teilweise viele Wochen lang) zu durchsuchen, um mögliche Herzprobleme zu erkennen. Die übliche Schreibgeschwindigkeit eines EKGs beträgt 50mm/s. Eine Aufzeichnung von einer Woche hätte dementsprechend eine Länge von 30,24 km. Damit daher ein Nutzen aus den Daten gezogen werden kann, müssen automatische und zuverlässige Algorithmen zur Erkennung von Herzanomalien entwickelt werden, welche die medizinischen Forscher*innen mit der Bewältigung eines meherere Gigabyte großen Datensatzes unterstützen.

1.2 ZIEL

Bei einer großen Datenmenge ist es wichtig, automatisierte und zuverlässige Wege für medizinische Forscher*innen zu schaffen, relevante Stellen für sie herauszufiltern. Damit die Algorithmen zum Finden der relevanten Stellen bestmöglich funktionieren und zudem eine bessere Lesbarkeit geschaffen wird, müssen die Daten zunächst von technischen Artefakten bereinigt werden. Ein technisches Artefakt wird über die Definition 1.2.1 innerhalb dieser Arbeit festgelegt:

Definition 1.2.1: Technisches Artefakt

Als technische Artefakte werden Vorkommnisse in den Daten definiert, bei denen die Ursache der vorliegenden Datenpunkt / des vorliegenden Datenpunkts auf technischer Seite liegt und daher die aufgezeichneten Daten nicht mit den echten Vitalwerten des Patienten übereinstimmen.

Insbesondere bei EKG-Daten kommt es sehr schnell zu technischen Artefakten, die unterschiedliche Ursachen haben können. Bewegt sich ein Patient, so führt dies zu einer Grundlinienwanderung, ein sogenanntes *Baseline Wander*. Ist keine ausreichende Erdung vorhanden oder liegt ein schlechter Elektrodenkontakt bzw. verschmutzte Elektroden vor, so führt dies zu regelmäßigen feinen Schwankungen des Signals. Dieses wird als *Power-Line-Interferenz* bezeichnet. Muskelzittern eines Patienten dagegen führt zu unregelmäßigen Ausschlägen und wird als *Muscle Artefact* beschrieben.

Die Datenmenge beinhaltet sowohl medizinisch unauffällige als auch medizinisch auffällige Abschnitte, die von gesteigertem Forschungsinteresse sein können. Die auffälligen Abschnitte werden innerhalb der Arbeit als medizinische Anomalie definiert, wie aus Definition 1.2.2 zu entnehmen ist.

Definition 1.2.2: Medizinische Anomalie

Medizinische Anomalien sind medizinisch auf Patientenseite begründete Anomalien, wodurch diese eine Abbildung der Vitalwerte des Patienten darstellen, jedoch diese nicht der normalen Wertestruktur des Parameters in Bezug auf einen gesunden Patienten entsprechen.

Anhand dieser beiden Definition lassen sich als Ziel dieser Arbeit zwei Forschungsfragen ableiten:

1. Welche Methode eignet sich zur automatisierten Elimination technischer Artefakte, um die ursprünglichen, realen Vitalwerte des Patienten wiederherzustellen? Dies hat die Hintergründe eine bessere Lesbarkeit zu schaffen und eine gute Vorverarbeitung der Daten zur Anomalieerkennung zu erbringen.

2. Wie können medizinische Anomalien zuverlässig und automatisiert mithilfe eines Klassifikationsmodells gefunden werden? Dies hat den Hintergrund, die große Datenmenge auf die Abschnitte zu reduzieren, welche für die weitere medizinische Forschung von Interesse sein können.

1.3 STRUKTUR

Die Masterarbeit ist in sechs Kapitel untergliedert. Im nächsten Kapitel (Kapitel 2) werden die theoretischen Grundlagen erläutert, die zum Verständnis der Masterarbeit notwendig sind. Innerhalb des Kapitels wird zunächst auf die medizinschen Grundlagen eingegangen, um ein Grundwissen zu einer EKG-Aufzeichnung zu schaffen. Daraufhin wird ein Überblick über das generelle Thema Anomalieerkennung gegeben. Die darauf folgenden Kapitel innerhalb der theoretischen Grundlagen beziehen sich speziell auf das Finden von medizinischen Anomalien in EKG-Daten. Dies lässt sich in die Elimination von technischen Artefakten, die Herzschlagerkennung und -segmentierung, die Feature-Extraktion und die Herzschlagklassifikation untergliedern. Jeder dieser Schritte findet sich in einem Unterkapitel wieder, in dem eine Methodenvorstellung über die eingesetzten und möglichen Methoden gegeben wird.

Nach Abschluss der Theorie werden die einzelnen Schritte auf die Charité-Daten unter Zuhilfenahme von öffentlichen Daten angewendet. Dieses Vorgehen ist ebenfalls in Abbildung 1.1 dargestellt.

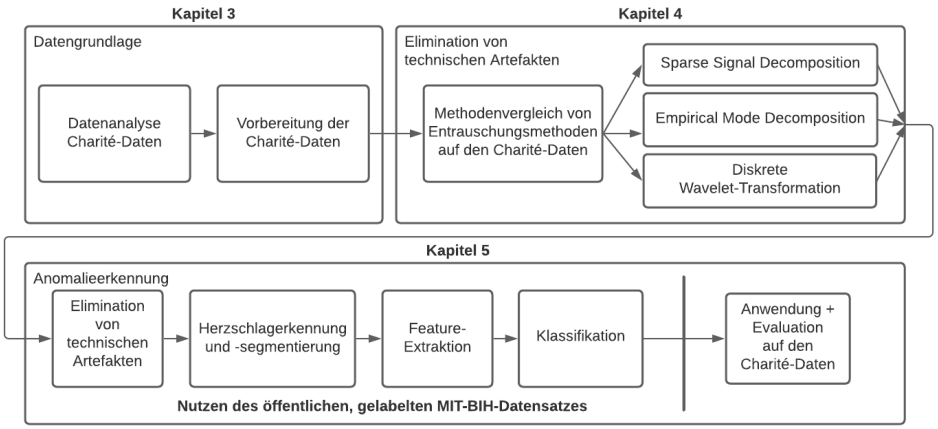

Abbildung 1.1: Abfolge der Umsetzungsschritte innerhalb der Masterarbeit
Quelle: Eigene Darstellung

Zunächst wird in Kapitel 3 die Datengrundlage vorgestellt, die das Ergebnis der Datenanalyse und der Vorverarbeitung der Charité-Daten beinhaltet.

Kapitel 4 geht daraufhin auf das Thema der Elimination von technischen Artefakten zur Beantwortung der ersten Forschungsfrage genauer ein, indem drei verschiedene Methoden auf Basis der Charité-Daten verglichen und evaluiert werden. Bei den drei Methoden handelt es sich um die Sparse Signal Decomposition, die Diskrete Wavelet-Transformation (DWT) und die Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN). Alle drei *Unsupervised*-Methoden zeigen in Studien eine gute Fähigkeit technische Artefakte zu eliminieren - bezogen auf eine andere Datengrundlage -, weshalb sie zur Evaluation evaluiert werden. Das Ziel des Vergleichs ist es, die Methoden mit dem bestmöglichen Ergebnis auf den Daten der Charité herauszufinden und zur weiteren Rauschentfernung einzusetzen.

Bei den Charité-Daten handelt es sich um ungelabelte Daten. Label werden lediglich indirekt über das Alarmsystem mitgegeben, wobei jedoch oft Falsch-Alarme auftreten. Studien zeigen, dass die Anomalieerkennung auf EKG-Daten insbesondere mithilfe des *Supervised*-Ansatzes funktioniert. Die einzigen Label, die in den Charité-Daten zur Verfügung stehen, sind jedoch die Alarme des Monitoring-Systems. Diese labeln Auffälligkeiten von EKG-Sequenzen, die jedoch oft unzuverlässig sind. Daher wird entsprechend der Prinzipien des *Transfer Learnings* dem Ansatz nachgegangen ein Modell auf öffentlichen Daten zu trainieren und das Modell auf einen manuell erstellten Teil-Datensatz der Charité-Daten zu übertragen. Als öffentliche Datenquelle wird der MIT-BIH-Datensatz verwendet, welcher sich als Standard-Testdatensatz zur Entwicklung von Anomalie-Detektoren etabliert hat [48]. Als Modell wird eine SVM eingesetzt, welche mithilfe von Hauptkomponenten auf Basis von DWT-Koeffizienten und unabhängigen Komponenten des EKG-Signals einzelne Herzschläge klassifiziert. Dieser Prozess wird durch

das Kapitel 5 beschrieben und entspricht der Beantwortung der zweiten Forschungsfrage.

Das Kapitel 6 zieht zum Schluss ein Fazit und gibt einen Ausblick auf mögliche Verbesserungen.

Zur besseren Übersichtlichkeit werden innerhalb der Arbeit Matritzen und Vektoren fett gedruckt dargestellt. Zudem wird eine Differenzierung von normalen und abnormalen Herzschlägen vorgenommen. Mit einem normalen Herzschlag wird im Weiteren ein Herzschlag bezeichnet, welcher der Wertestruktur eines gesunden Menschen entspricht. Liegt eine Abweichung von dieser Wertestruktur vor, so wird der Herzschlag als abnormal bezeichnet. Der Funktionsparameter in den Abbildungen der EKG-Zeitreihen, welcher den aufgezeichneten Werten entspricht, wird einheitlich als $Y(x)$ definiert.

1.4 TECHNISCHE INFRASTRUKTUR

Die technische Grundlage zur Analyse der Daten bildet aufgrund der hohen Datenmenge Apache Spark in Verbindung mit Hadoop. Hadoop wird als Speichertechnologie eingesetzt, die für große Datenmengen optimiert ist. Apache Spark entspricht einer verteilten In-Memory-Datenverarbeitungsengine, mit welcher Daten parallel und verteilt in einem Cluster von mehreren Rechnern gleichzeitig verarbeitet werden können [55].

Durch die verteilte Verarbeitung ergeben sich Herausforderungen an die zu entwickelnden Verarbeitungsprozesse, da die Daten nicht vollständig zu einem Zeitpunkt im Arbeitsspeicher gehalten werden können. Dies erfordert darauf ausgelegte Konzepte, welche in dieser Arbeit berücksichtigt und entwickelt werden müssen.

Zur Entwicklung der Spark-Anwendung wird auf die Scala- und Python-API zurückgegriffen. Da Spark in Scala geschrieben ist, ist die Scala-API stets aktuell. Zusätzlich bietet die Scala-API den Vorteil, dass keine zusätzlichen Interpreter-Prozesse mit der Java Virtual Machine (JVM) kommunizieren müssen, wie es bei Python der Fall ist. Aus diesem Grund bestehen signifikante Performanceunterschiede zwischen der Scala- und der Python-API, vor allem bei der Verwendung der User-Defined Functions (UDFs). In Python gibt es dafür beispielsweise eine Großzahl an Visualisierungsbibliotheken, weshalb in der Masterarbeit je nach Anwendungsfall auf die entsprechend geeignetere API zurückgegriffen wird.

2 THEORETISCHE GRUNDLAGEN

In diesem Kapitel werden medizinische Grundlagen, Grundlagen zum Thema Anomalieerkennung sowie grundlegende Prinzipien und Methoden zur Elimination technischer Artefakte und zur Findung medizinischer Anomalien vorgestellt.

2.1 MEDIZINISCHE GRUNDLAGEN

Das EKG ist ein weit verbreitetes Mittel zur Diagnose von Herzerkrankungen und beschreibt eine nicht invasive Untersuchungsmethode zur Erfassung der Herzaktion und Lage des Herzens [77]. Der Parameter zeigt eine graphische Darstellung der Ausbreitung der elektrischen Erregung von Vorhof und Ventrikelmyokard, gemessen durch die Veränderung der Potentialdifferenz zwischen zwei Punkten an der Körperoberfläche, welche gegen die Zeit aufgezeichnet wird [26]. Damit die einzelnen Komponenten des typischen EKG-Verlaufs nachvollziehbar werden, wird im ersten Schritt auf die Elektrophysiologie mit dem Fokus auf der Ausbreitung des elektischen Signals im Herzen eingegangen. Im zweiten Schritt wird daraufhin gezielt die Gestalt einer EKG-Zeitreihe erläutert.

Elektrophysiologie des Herzens

Die Aufgabe des Herzens ist es, den Körper mit sauerstoffreichem Blut zu versorgen. Dafür durchläuft das Blut wiederkehrend denselben Kreislauf, dessen zentrale Instanz das Herz bildet. Innerhalb des Kreislaufs kommt das Blut über die Venen am rechten Vorhof an und wird weiter in die rechte Herzkammer geleitet. Von dort aus gelangt es zur Anreicherung mit Sauerstoff in die Lunge und fließt über den linken Vorhof in die linke Herzkammer. Durch eine Pumpbewegung wird das nun angereicherte Blut zu den Organen, der Muskulatur und dem Gehirn verteilt. Nach Verbrauch des Sauerstoffs strömt das sauerstoffarme Blut wieder zum Herzen zurück und der Kreislauf beginnt erneut. Die Pumpbewegung wird als eine rhythmische Kontraktion durchgeführt. Dabei wird die Anspannungsphase als Systole und die Erschlaffungsphase als Diastole bezeichnet. Bei der Systole zieht sich der Herzmuskel zusammmen, wodurch das Blut in den Lungenkreislauf und den Körper gepumpt wird. Bei der Diastole erschlafft der Muskel und die Herzkammern werden wieder mit Blut gefüllt. Die Systole erfolgt über elektrische Erregungen, dem sogenannten Reizleitungssystem. Die relevanten Komponenten des Reizleitungssystems werden in Abbildung 2.1 dargestellt.

Die primäre Reizbildung findet im Sinusknoten statt, welcher sich an der Muskulatur des rechten Vorhofs an der Einmündungsstelle der Vena cava

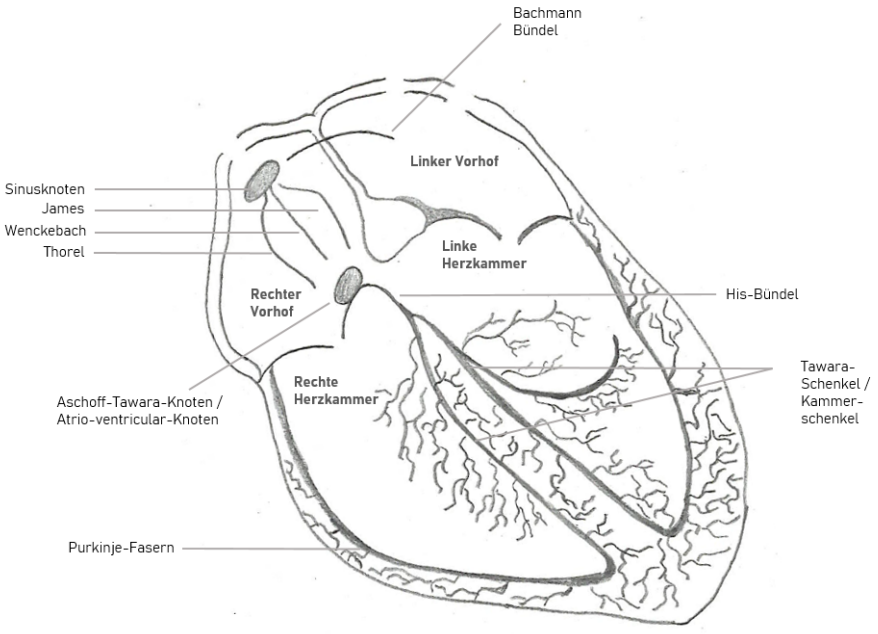

Abbildung 2.1: Relevante Komponenten des Reizleitungssystems
Quelle: In Anlehnung an Borchert, 2005 [7])

befindet. Von dort aus wird der Reiz über spezifisches Muskelreizleitungsgewebe weitergeleitet. Das Bachmann-Bündel dient der Anregung des linken Vorhofs. Der rechte Vorhof wird über die Bündel James, Wenckebach und Thorel innerviert. Die drei zuletzt genannten Muskelreizleitungsgewebe sorgen zudem für die Reizübertragung an den Aschoff-Tawara-Knoten. Aus diesem gehen Reizleitungsfasern für das His-Bündel ab, welches den Reiz in die Herzkammern überführt. Diese Reizweiterleitung erfolgt über die Tawara-Schenkel an die Purkinje-Fasern, welche wiederum das Arbeitsmyokard erregen und zu einer koordinierten Kontraktion führt.

Messung und Gestalt des Elektrokardiogramms

Der Verlauf der Anspannungs- und Erschlaffungsphase kann anhand eines Elektrokardiogramms nachvollzogen werden. Grund dafür ist ein elektrisches Feld, welches durch die elektrischen Impulse im Herzen erzeugt wird. Mittels Elektroden, die an verschiedenen Körperregionen angebracht werden, sind daher Spannungsunterschiede zu beobachten. Dabei gibt es unterschiedliche Ansätze, wie die elektrischen EKG-Feldpotenziale abgeleitet werden. Standardmäßig wird ein 12-Kanal-EKG angewendet. Zur Aufzeichnung der zwölf Ableitungen, werden vier Elektroden an den jeweiligen Extremitäten und sechs Elektroden am Brustkorb befestigt. Die Ableitungen setzen sich zusammen aus den Einthoven-, Goldberger- und Wilson-Ableitungen. Die Ableitungen nach Einthoven (Tabelle 2.1) sind bipolare Extremitätenableitungen, welche die Potenzialdifferenz zwischen zwei gleichberechtigten Elektroden messen [77].

Ableitung	Elektrodenposition
I	rechter Arm (negativ) / linker Arm (positiv)
II	rechter Arm (negativ) / linkes Bein (positiv)
III	linker Arm (negativ) / linkes Bein (positiv)

Tabelle 2.1: Extremitätenableitungen nach Einthoven

Die vierte Elektrode (rechtes Bein) dient der Erdung. Die Ableitungen nach Goldberger (Tabelle 2.2) entsprechen unipolaren Extremitätenableitungen, welche jeweils zwei der Extremitätenelektroden über hochohmige Widerstände zu einer indifferenten Elektrode zusammenschaltet und die Potenzialänderung an der verbliebenen differenten Elektrode misst [77].

Ableitung	Elektrodenposition
aVR	rechter Arm (positiv) / linker Arm und linkes Bein (negativ)
aVL	linker Arm (positiv) / rechter Arm und linkes Bein (negativ)
aVF	linkes Bein (positiv) / linker und rechter Arm (negativ)

Tabelle 2.2: Extremitätenableitungen nach Goldberger

Bei den Ableitungen nach Wilson wird die Potenzialdifferenz zwischen den Brustwandelektroden und einer indifferenten Sammelelektrode, welche durch einen Zusammenschluss der Extremitätenableitungen gebildet wird, abgeleitet [77]. Sie werden als V1 bis V6 bezeichnet und dienen der Überwachung der Erregungsausbreitung in der Horizontalebene. Die typischen Charakteristika der EKG-Ableitungen können anhand eines Sinusrhythmus erläutert werden, welcher in Abbildung 2.2 dargestellt wird.

Die P-Welle ist auf die Erregung der Vorhöfe zurückzuführen. Die PQ-Strecke entspricht der Kammerüberleitungszeit. Der QRS-Komplex beschreibt die Kammererregung, die ST-Strecke die Kontraktion der Kammern und die T-Welle die Repolarisation. Die kleine positive Welle hinter der T-Welle wird als U-Welle bezeichnet. Sie zählt mit zu den Erregungsrückbildungsstörungen [7]. Die verschiedenen Intervalle sind insbesondere für die zu erkennende Regelmäßigkeit der Charakteristika relevant.

Technische Artefakte im EKG

Bei einer kontinuierlichen EKG-Messung kann das Signal regelmäßig durch verschiedenes Rauschen verfälscht werden [34]. Typisches Rauschen bei EKG-Messungen sind Baseline Wander (BW), Power-Line-Interferenz (PLI), Muscle Artefacts (MA) und Additive White Gaussian Noise (AWGN) [34]. Abbildung 2.3 zeigt Beispiele für die vier verschiedenen Arten von Rauschen.

Die einzelnen Artefakte begründen sich aufgrund verschiedener Ursachen. Baseline Wander tritt auf, wenn sich der Patient bewegt oder die Elektrode selbst bewegt wird [63]. Muscle Artefacts entwickeln sich - wie der Name

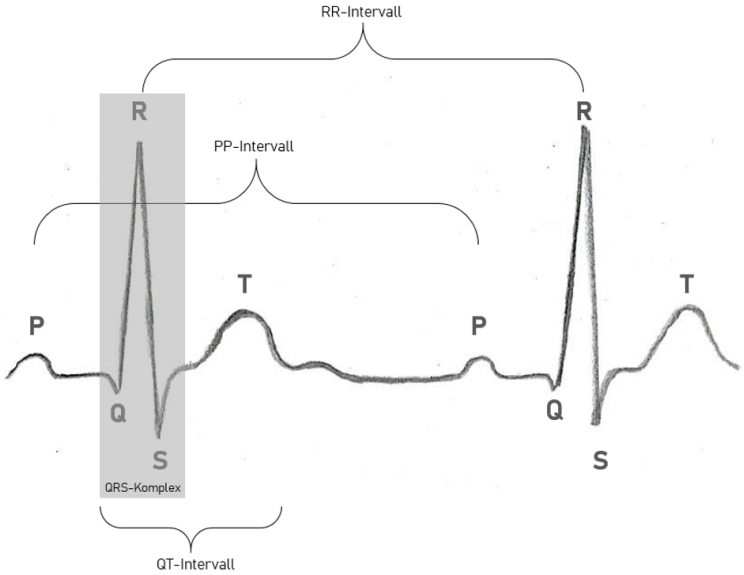

Abbildung 2.2: Normales EKG-Signal mit entsprechender Notation
Quelle: In Anlehnung an Li et al., 2020 [37]

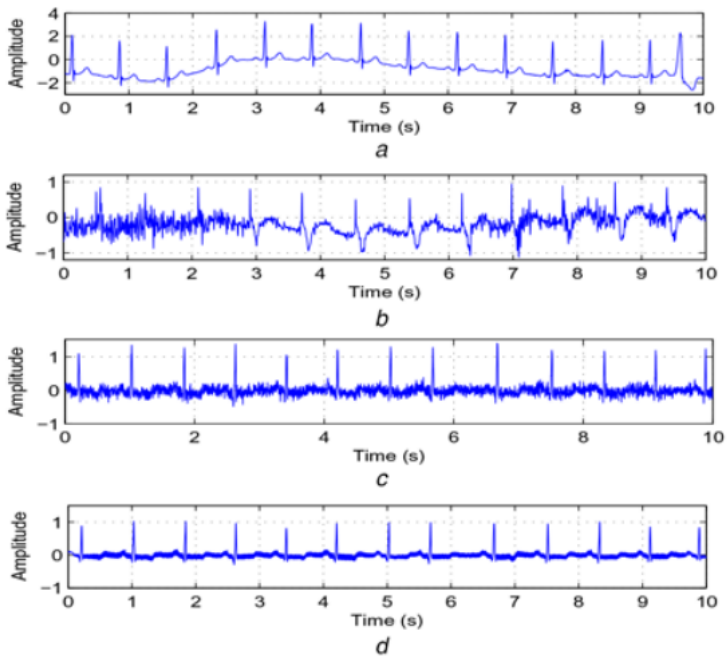

Abbildung 2.3: EKG-Signal mit Baseline Wander a), Muscle Artefact b), Additive White Gaussian Noise c) und Power-Line-Interferenz d)
Quelle: Kumar et al., 2020 [34]

schon sagt - aufgrund von Muskelzittern, beispielsweise aufgrund von Kälte. Additive White Gaussian Noise basiert auf schlechten Bedinungen der Kanalübertragung. Es kann alle Frequenzkomponenten enthalten und wird auch Kanalrauschen genannt [71]. Power-Line-Interferenz entsteht aufgrund

unzureichender Erdung, schlechtem Elektrodenkontakt oder verschmutzten Elektroden [63].

2.2 ANOMALIEERKENNUNG

Die Anomalieerkennung wird seit vielen Jahren in Bezug auf eine große Anzahl verschiedener Anwendungsgebiete wissenschaftlich ausgearbeitet [3, 12, 24, 30] und beschreibt das Finden von Mustern in Daten, die nicht mit dem erwarteten Verhalten übereinstimmen [12]. In diesem Zusammenhang werden die Begriffe Neuheitenerkennung, Ausreißererkennung und Anomalieerkennung oft als Synonyme verwendet. Auch die Disziplin der Rauschentfernung wird regelmäßig mit der Anomalieerkennung in Relation gesetzt. Jedoch ist allen Begriffen eine andere Bedeutung zuzuschreiben, weshalb sie an dieser Stelle nach Chandola et al. [12] abgegrenzt werden: Rauschen bezeichnet Vorkommnisse in den Daten, die nicht von Interesse sind, jedoch die Datenanalyse behindern. Die Anomalieerkennung hingegen befasst sich mit dem Finden von Mustern in den Daten, die eine thematische Relevanz aufweisen. Die unerwünschten Störungen in den Daten werden daher im Zuge der Rauschentfernung vor der Datenanalyse eliminiert. Die Neuheitenerkennung beschreibt das Erkennen von zuvor unbeobachteten Mustern in den Daten. Hier ist die Abgrenzung durch den Umstand gegeben, dass die gefundenen Muster nach der Erkennung, im Gegensatz zur Anomalieerkennung, in das normale Modell mit aufgenommen werden.

Die Ausreißererkennung beschreibt eine Teildisziplin der Anomalieerkennung. Hawkins [28] definierte einen Ausreißer wie folgt:

> „ An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.“

Aggarwal [2] bezeichnet im Zusammenhang mit der Definition eines Ausreißers diesen auch als Anomalie auf Basis der Data Mining und Statistikliteratur. Die Definition einer Anomalie deckt dagegen deutlich mehr ab, als die von Hawkins oder Aggarwal aufgestellte Definition eines Ausreißers. Chandola [12] unterteilt eine Anomalie in drei Subgruppen mit voneinander differenzierbaren Definitionen, welche im nachfolgenden Kapitel 2.2.1 genauer beschrieben werden.

2.2.1 *Anomaliearten*

Anomalien beschreiben Abweichungen einzelner Datenpunkte oder einer Gruppe von Datenpunkten, die nicht mit dem normalen, erwarteten Verhalten der Daten übereinstimmen [12]. Diese Abweichung von der Norm kann unterschiedliche Gründe haben, wie beispielsweise Kreditkartenbetrug, Systemausfälle oder medizinische Vorfälle beim Patienten. Alle Anomalien haben jedoch die Gemeinsamkeit, dass sie interessant und relevant für den Analysten sind und daher mittlerweile eine Vielzahl von Methoden zur Er-

kennung von Anomalien erforscht wurden [12, 24]. Bei Chandola et al. [12] werden Anomalien in drei Kategorien unterschieden.

Punktanomalie

Weicht ein einzelner Datenpunkt von dem Rest der Daten ab, so liegt eine Punktanomalie vor. Diese Kategorie ist die einfachste Art der Anomalien und ist am ehesten mit der Definition eines Ausreißers in Verbindung zu bringen. Die Grafik 2.4 zeigt ein solches Beispiel. Abgebildet werden die Transaktionssummen über einen Zeitraum. Eine Transaktionssumme weicht sehr stark von den übrigen Transaktionssummen ab. Diese Beobachtung wird als Punktanomalie bezeichnet.

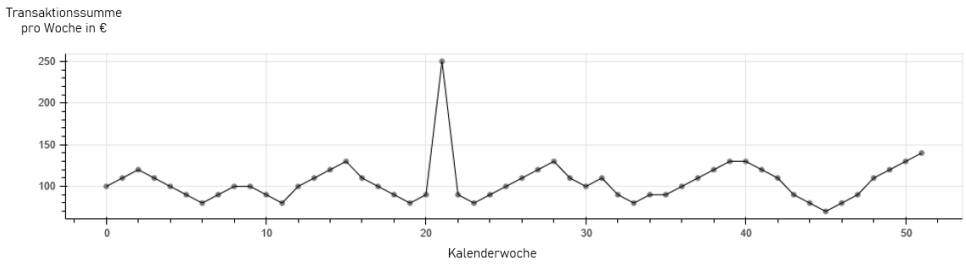

Abbildung 2.4: Beispiel für eine Punktanomalie
Quelle: In Anlehnung an Chandola et al., 2009 [12]

Kontextuelle Anomalie

Ist ein Punkt in einem spezifischen Kontext abnormal, aber nicht wenn dieser Kontext nicht gegeben ist, dann liegt eine kontextuelle Anomalie vor. Der Kontext wird dabei durch die Struktur der Daten selbst induziert, indem immer mindestens ein kontextuelles Attribut und mindestens ein Verhaltensattribut vorliegt. Die kontextuellen Attribute bestimmen den Kontext, wie das Zeitattribut in Zeitreihen. Die Verhaltensattribute definieren die nichtkontextuellen Charakteristika eines Datenpunktes, anhand derer im Kontext ein normales oder abnormales Verhalten definiert werden kann. Diese Art von Anomalie kommt insbesondere in Zeitreihendaten und räumlichen Daten vor. Die Grafik 2.5 zeigt ein Beispiel für diese Anomaliekategorie. Gezeigt wird hier eine Ein-Jahres-Zeitreihe über den durchschnittlichen Neuschnee pro Woche in Zentimetern. In Woche 23 liegt der Durchschnitt (Punkt p1) bei 15cm. Dieser Wert kommt auch zu anderen Zeitpunkten vor (Punkt p2), jedoch wird durch das Zeitattribut der Kontext induziert, dass es sich bei der Woche 23 um eine Woche Anfang Juli handelt und es sich daher in Bezug auf den jährlichen Schneefallverlauf um eine abnormale Beobachtung handelt.

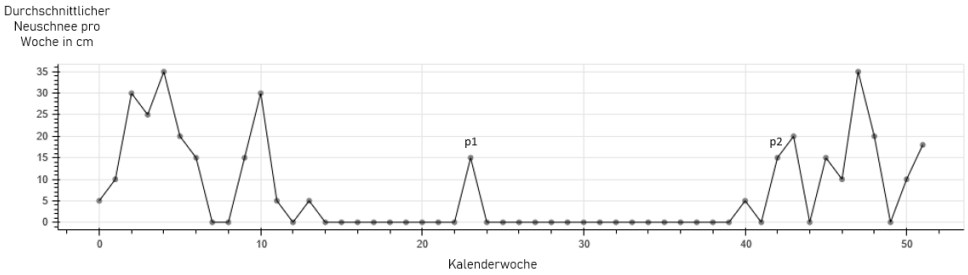

Abbildung 2.5: Beispiel für eine kontextuelle Anomalie
Quelle: Eigene Darstellung

Kollektive Anomalie

Eine kollektive Anomalie bezeichnet eine Kollektion verwandter Datenpunkte, die in Bezug auf den gesamten Datensatz abnormal sind. Dabei können einzelne Datenpunkte innerhalb der Kollektion das erwartete Verhalten aufweisen, das gemeinsame Auftreten der Datenpunkte ist jedoch abormal. Ein Beispiel für diese Kategorie wird durch Abbildung 2.6 gegeben. Hier wird die erste EKG-Ableitung nach Einthoven dargestellt. Alleine gesehen wären die grau markierten Punkte in keinem abnormalen Wertebereich. Durch das Auftreten der Punkte in der Kollektion führt es jedoch zu einem unregelmäßigen Herzrhythmus, welcher eine Anomalie darstellt.

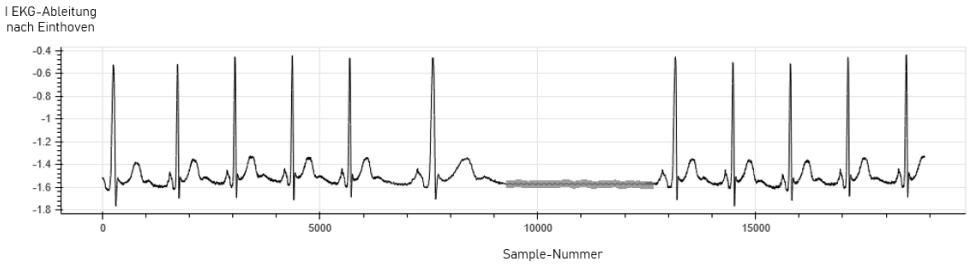

Abbildung 2.6: Beispiel für eine kollektive Anomalie
Quelle: In Anlehnung an Chandola et al., 2009 [12]

2.2.2 *Anomalieerkennungsstrategien*

Das Erkennen von Anomalien kann über verschiedene Ansätze geschehen und hängt von den vorliegenden Daten ab. Es wird dabei zwischen *Supervised-*, *Semi-Supervised-* und *Unsupervised*-Modellen unterschieden, je nachdem, ob Label vorliegen [12, 20]. Die Label geben an, ob es sich um eine normale oder abnormale Struktur handelt. Gelabelte Daten zur Erkennung von Anomalien, welche für alle möglichen Anomalie-Ausprägungen repräsentativ sind, zu erhalten, stellt oft eine Herausforderung dar. Insbesondere die Kennzeichnung der abnormalen Strukturen erfordert meistenfalls manuelles Labeln durch einen Experten, was sehr viel Zeit und Wissen erfordert [12]. Zudem können sich Anomalien auch im Laufe der Zeit verändern, wo-

durch ein vorherig gelabelter Datensatz nicht mehr repräsentativ ist. Einige Anomalien sind auch nicht eindeutig zu identifizieren, da nicht immer eindeutig definiert ist, wie sich eine Anomalie in den Daten äußert. Zusätzlich können bestimmte Vorabinformationen über Eigenschaften der Anomalien dafür sogen, dass ein gelabelter Datensatz nicht notwendig ist. Aus diesem Grund ist die Auswahl der richtigen Strategie zur Anomalieerkennung essentiell für gute Endresultate. Die drei verschiedenen Anomalieerkennungsstrategien werden im Folgenden kurz erläutert.

Supervised-Anomalieerkennung

Bei der Supervised-Anomalieerkennung wird ein vollständig gelabelter Datensatz vorausgesetzt. Das bedeutet, dass sowohl die normalen als auch die abnormalen Strukturen gekennzeichnet sein müssen. Ein Hauptproblem des Supervised-Ansatzes bildet die geringere Menge an Anomalien verglichen mit der Menge an Daten in einem normalen Wertebereich. Das Problem basiert auf dem Thema der unbalancierten Klassenverteilung und muss in den Modellen berücksichtigt werden. Des Weiteren besteht oftmals eine Schwierigkeit darin, für die Anomalien repräsentative Label zu finden, mit denen die Supervised-Modelle umgehen können [12].

Semi-Supervised-Anomalieerkennung

Die Semi-Supervised-Anomalieerkennung kann in zwei Richtungen interpretiert werden. Zum einen kann sie eine Strategie bezeichnen, welche Modelle trainiert, die einen Datensatz voraussetzen, bei dem nur die normalen Strukturen gelabelt sind [12]. Durch diesen Ansatz wird dem zuvor beschriebenen Problem der Bezeichnungsfindung im Supervised-Ansatz aus dem Weg gegangen. Zum anderen bezeichnet es die Möglichkeit Anomalien mit Datensätzen zu finden, bei denen lediglich ein kleiner Teil der Daten gelabelt ist [59]. Bei den gelabelten Daten handelt es sich sowohl um Anomalien als auch um normale Strukturen. Dieser Ansatz ist insbesondere im Deep Learning vertreten [59].

Unsupervised-Anomalieerkennung

Unsupervised-Ansätze beschreiben Modelle, die keine Label zur Klassifikation von normalen und abnormalen Strukturen benötigen, sondern auf Basis der Daten selbst abnormale Datenpunkte finden. Durch die nicht benötigten Label sind sie auf alle Datensätze anzuwenden und entsprechen daher auch einem verbreiteten Vorgehen zur Anomalieerkennung [12].

2.3 METHODEN ZUR ELIMINATION TECHNISCHER ARTEFAKTE

Wie durch die Abbildung 2.6 in Kapitel 2.2.1 bereits gezeigt wurde, sind Veränderungen der Struktur eines EKGs in die Kategorie der kollektiven Anomalie einzuordnen. Damit einzelne Anomalieerkennungsstrategien zum

Finden dieser kollektiven Anomalien auf EKG-Signale angewendet werden können, sind einige Preprocessing-Schritte notwendig. Diese werden in diesem und den weiteren Kapiteln genauer beschrieben. Zunächst ist die Bereinigung der Daten von technischen Artefakten durchzuführen. Die in EKG-Daten vorkommenden technischen Artefakte werden auch dem Oberbegriff des Rauschens zugeordnet. Wie in Kapitel 2.2 bereits erläutert wurde, steht die Rauschentfernung stark mit der Anomalieerkennung in Verbindung, da sie oftmals einen notwendigen Schritt zur Vorbereitung der Daten für die Anomalieerkennung darstellt. In diesem Zuge können Störungen in den Daten beseitigt werden, sodass sie das Auffinden von Anomalien nicht negativ beeinflussen. Für Zeitreihen existieren verschiedene Ansätze, um das Rauschen aus den Daten zu eliminieren. Auf Basis der Literatur werden diverse Methoden für die Elimination technischer Artefakte in EKG-Daten regelmäßig eingesetzt. Dazu zählen unter anderem digitale Filter, wie Hochpass-, Tiefpass-, Band-pass- und Notchfilter [37]. Oftmals wird eine Kombination aus Hochpass- und Tiefpassfilter gewählt, um korrespondierendes Rauschen entfernen zu können [37]. Eine weitere Möglichkeit ist die Nutzung von adaptiven Filtern, wie der Least Mean Square (LMS) und der Recursive Least Square (RLS) [37]. Insbesondere der Normalized Least Mean Square (NLMS) als Erweiterung des LMS konnte sich bei Saxena et al. [63] im Vergleich zu anderen Entrauschungsmethoden von diesen absetzen.

Die DWT stellt eine weitere oft genutzte Methode dar [37, 50, 66]. Die wichtigste Aufgabe bei der Entrauschung mittels DWT ist die richtige Wahl der zugrunde liegenden Basisfunktion, da je nach Basisfunktion unterschiedliche Ergebnisse resultieren [37]. Eine bisher eher selten eingesetzte Methode ist die Sparse Signal Decomposition. Sie führte jedoch bei Satija et al. [61] und Kumar und Sharma [34] mit einer durchschnittlichen Erkennungsgenauigkeit von 99% zu sehr guten Ergebnissen. Als letztes ist die Bereinigung mittels der Empirical Mode Decomposition (EMD), beziehungsweise der Erweiterung CEEMDAN, zu erwähnen, welche sich ebenfalls auf EKG-Daten bewährt haben [73].

Die drei Methoden Sparse Signal Decomposition, DWT und CEEMDAN werden im Weiteren genauer beleuchtet.

2.3.1 *Sparse Signal Decomposition*

Die Sparse Signal Decomposition ist eine Methode, die von Satija et al. [61] 2016 erstmals für die Anwendung auf EKG-Daten publiziert wurde. Die Methode ist eine Anwendung des Dictionary Learnings, welches eine Repräsentationsmethode beschreibt, die mithilfe einer übervollständigen Basis (im Folgenden Dictionary) und einem der Basis angenäherten, spärlich besetzten Vektors (Sparse-Vektor), ein Signal möglichst gut repräsentieren soll [17, 56].

Bei der Sparse Signal Decomposition wird - ähnlich zum Grundgedanken der Fourier-Transformation - dem Ansatz nachgegangen, dass jedes Signal durch die Summe verschiedener Basisfunktionen, wie beispielsweise Sinus- und Kosinuswellen, dargestellt werden kann. Die folgende Metho-

denbeschreibung basiert auf den publizierten Papern von Satija et al. aus dem Jahr 2016 [61] und 2017 [60, 62], wie auch auf dem aufsetzenden Paper von Kumar und Sharma aus dem Jahr 2020 [34]. Die Beschreibung ist speziell auf die Anwendung auf EKG-Daten angepasst, kann jedoch auch generell für das Entrauschen von Zeitreihendaten genutzt werden. Zur besseren Übersichtlichkeit werden die einzelnen Schritte der Sparse Signal Decomposition in Unterkapitel gegliedert. Dabei wird zunächst auf die Zieldarstellung der Zeitreihen-Repräsentation eingegangen, welche die Grundlage für die Entrauschungsprozedur bildet. Daraufhin wird die Erstellung des Dictionaries erläutert und die darauf basierende Berechnung des Sparse-Vektors beschrieben. Zum Schluss wird auf den Eliminationsprozess mithilfe der Sparse Signal Decomposition eingegangen.

Sparse Repräsentation

Jede Zeitreihe $\mathbf{x} = (x_1, x_2, ..., x_P)^T$ kann mit zeitlokalisierten und frequenzlokalisierten Elementarwellenformen durch eine Matrix $\mathbf{\Psi} \in \mathbb{R}^{PxQ}$ und einen Sparse-Vektor $\boldsymbol{\alpha} \in \mathbb{R}^Q$ dargestellt werden, wobei $P < Q$ als Restriktion gegeben ist:

$$\mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha} = \sum_{i=1}^{Q} \alpha_i \boldsymbol{\psi_i} \tag{2.1}$$

$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_Q]$ bezeichnet den Sparse-Vektor, der die Koeffizienten für die frequenzlokalisierten Elementarwellenformen aus der Matrix $\mathbf{\Psi}$ liefert. $\boldsymbol{\psi_i}$ gibt die i-te Spalte der Matrix $\mathbf{\Psi}$ an. P beschreibt die Länge der Zeitreihe $\mathbf{x}$. Q entspricht der Anzahl an genutzten Elementarwellenformen zur Repräsentation von $\mathbf{x}$. Die Matrix $\mathbf{\Psi}$ wird auch als Overcomplete Dictionary bezeichnet.

Design des Overcomplete Dictionary

Das Overcomplete Dictionary kann durch die Verwendung grundlegender Wellenformen aus verschiedenen Analysefunktionen erstellt oder adaptiv über einen Trainingsdatensatz gelernt werden. Als Analysefunktion wird die Basisfunktion, wie beispielsweise eine Sinus- oder Kosinusfunktion, bezeichnet, auf die mindestens ein Teil der Wellenformen im Dictionary basieren. In den Papern zur Anwendung der Methode auf EKG-Daten wird auf den Ansatz der Verwendung grundlegender Wellenformen zurückgegriffen, da Vorwissen über die zeitlichen und spektralen Eigenschaften des EKG-Signals und verschiedener technischer Artefakte besteht [34, 61].

Ziel ist es die technischen Artefakte Baseline Wander, Power-Line-Interferenz, Muscle Artifacts und additives weißes gaußsches Rauschen zu eliminieren. Die Artefakte können sowohl alleinstehend vorkommen als auch gemeinsam das Originalsignal verrauschen, indem sie additive und stetige Funktionen im verrauschten Signal darstellen. Zur Elimination wird zwischen frequenzlokalisierten und zeitlokalisierten Komponenten unterschieden. Frequenzlokalisierte Komponenten sind Komponenten, die aufgrund ihres Frequenzbereiches erkannt werden können. Sie beschreiben die nieder-

frequenten Bereiche im QRS-Komplex, die T-Welle und die P-Welle. Von den technischen Artefakten gehören Baseline Wander und Power-Line-Interferenz ebenfalls dazu. Die zeitlokalisierten, spitzartigen Komponenten sind die hochfrequenten Bereiche des QRS-Komplexes und Muscle Artifacts oder additives weißes gaußsches Rauschen aus den möglicherweise auftretenden technischen Artefakten. Daher besteht das Dictionary bei EKG-Daten aus einer vordefinierten Matrix der Größe PxQ und wird beschrieben als

$$\mathbf{\Psi} = [\mathbf{\Psi}_{BW}|\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]. \tag{2.2}$$

$\mathbf{\Psi}_{BW}$ beschreibt die Submatrix, die aufgrund ihres Frequenzbereiches die Struktur des Baseline Wanders modellieren kann. Baseline Wander liegt zwischen null und einem Hertz. $\mathbf{\Psi}_{LF}$ entspricht der Submatrix zur Modellierung der niederfrequenten Komponenten, wie der P-Welle, T-Welle und niederfrequenten Bereichen des QRS-Komplexes. Sie bewegen sich je nach Literatur zwischen einem und sechs Hertz [34], beziehungsweise einem und 20 Hertz [62]. $\mathbf{\Psi}_{PLI}$ dient der Modellierung der Struktur der Power Line Interferenz im Frequenzbereich zwischen 47 und 53 Hertz. Alle drei genannten Komponenten von $\mathbf{\Psi}$ können effektiv durch sinusförmige Wellen modelliert werden. $\mathbf{\Psi}_{HF}$ wird genutzt, um sowohl die hochfrequenten Bereiche des QRS-Komplexes (inklusive des R-Peaks), Muscle Artifacts als auch additives weißes gaußsches Rauschen zu erfassen. Da es sich um zeitlokalisierte Komponenten handelt, wird hier auf eine Einheitsmatrix der Größe PxP zurückgegriffen, sodass der dazugehörige Sparse-Vektor die Zeitreihe selbst repräsentiert und die zeitabhängigen Spitzen und impulsartigen Komponenten nicht verloren gehen. $\mathbf{\Psi}_{HF}$ wird auch als *impulse dictionary* bezeichnet und wird gesetzt als:

$$\mathbf{\Psi}_{HF} := \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{2.3}$$

Zur Modellierung der Submatritzen $\mathbf{\Psi}_{BW}$, $\mathbf{\Psi}_{LF}$ und $\mathbf{\Psi}_{PLI}$ wird auf das diskrete Fourier-Transformations-Dictionary zurückgegriffen, da es sich um cyclostationäre Merkmale handelt. Die Matrix der Diskreten Fourier-Transformation[1] (DFT) kann nach Satija et al. [60] als Kombination der eindimensionalen diskreten Sinus- und Kosinus-Transformation[2] betrachtet und geschrieben werden als:

$$\mathbf{DFT} = [\mathbf{S}|\mathbf{C}] \tag{2.4}$$

1 Weiterführende Informationen zur Diskreten Fourier-Transformation sind [43] und [8] zu entnehmen.

2 Weiterführende Informationen zur Diskreten Kosinus-Transformation sind [42] zu entnehmen.

Die einzelnen Koeffizienten von $\mathbf{S}_{ij}$ und $\mathbf{C}_{ij}$ werden berechnet durch:

$$[\mathbf{S}]_{ij} = \sqrt{\frac{2}{P}} \left[a_i \sin\left(\frac{\pi(2j+1)(i+1)}{2P} \right) \right], \tag{2.5}$$

$$[\mathbf{C}]_{ij} = \sqrt{\frac{2}{P}} \left[a_i \cos\left(\frac{\pi(2j+1)i}{2P} \right) \right], \tag{2.6}$$

wobei a_i definiert wird durch:

$$a_i := \begin{cases} \sqrt{\frac{1}{2}} & \text{, falls } i = P-1 \text{ in } \mathbf{S} \text{ oder } i = 0 \text{ in } \mathbf{C} \\ 1 & \text{, sonst.} \end{cases} \tag{2.7}$$

$$\text{mit } i, j = 0, 1, 2, ..., P-1$$

Die diskreten Sinus- und Kosinus-Transformationen (DST und DKT) werden anstelle der DFT verwendet, da sie komplexe Terme zur Lösung des im nächsten Unterkapitel beschriebenen Optimierungsproblems vermeiden. Beide Transformationen zerlegen einen zeitdiskreten Vektor endlicher Länge in eine Summe skalierter und verschobener Basisfunktionen. Jedoch wird in der Art der Basisfunktion unterschieden [42]. Die DFT verwendet eine Reihe harmonisch zusammenhängender Exponentialfunktionen, während die DST und DKT nur reelle Sinus- und Kosinusfunktionen nutzen. Die DST, beziehungsweise die DKT, können auch alleinstehend die Basisfunktionen bilden, da sie ebenfalls eine sparse Repräsentation liefern. Forschungsergebnisse zeigen, dass die Kombination beider Transformationen, insbesondere an den Grenzen, zu einem geringeren Rekonstruktionsfehler führt. Die Grenzen beschreiben den Anfang und das Ende einer Zeitreihe.

Die auf Basis der DST und DKT berechneten Matritzen $\mathbf{S}$ und $\mathbf{C}$ bilden die Grundlage für die Dictionaries $\mathbf{\Psi}_{BW}$, $\mathbf{\Psi}_{LF}$ und $\mathbf{\Psi}_{PLI}$. Auf Basis der Gleichung

$$\text{Spaltennummer in } \mathbf{S} \text{ und } \mathbf{C} \; = \lfloor \frac{2 \cdot P \cdot f}{f_s} \rfloor \tag{2.8}$$

kann berechnet werden, welche Spalten aus $\mathbf{S}$ und $\mathbf{C}$ die gesuchten Frequenzbereiche abdecken. f ist die Frequenz und f_s entspricht der Sampling-Rate. Die Sampling-Rate ist die Anzahl an Werten, die pro Sekunde aufgezeichnet werden. Wird beispielsweise das Dictionary $\mathbf{\Psi}_{BW}$ für ein Signal der Länge von 10 Sekunden gesucht, so wird $f = 0$ für die untere Grenze und $f = 1$ für die obere Grenze gesetzt. P kann durch $10 \cdot f_s$ substituiert werden.

$$\begin{aligned} \text{Spaltennummer untere Grenze} &= \lfloor 2 \cdot 10 sek. \cdot 0 Hz \rfloor \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{Spaltennummer obere Grenze} &= \lfloor 2 \cdot 10 sek. \cdot 1 Hz \rfloor \\ &= 20 \end{aligned}$$

Auf Basis dessen wird für dieses Signal $\mathbf{\Psi}_{BW} = [\mathbf{S}_{i,0-20}|\mathbf{C}_{i,0-20}]$ definiert. Durch ein simultanes Vorgehen werden die Spalten zum Erstellen der Dictionaries von $\mathbf{\Psi}_{LF}$ und $\mathbf{\Psi}_{PLI}$ gefunden, indem die entsprechenden Frequenzober- und -untergrenzen in die Formel zur Berechnung der Spalten eingesetzt wird.

Optimierungsproblem: Berechnung des Sparse-Vektors

Die Sparse-Koeffizienten $\boldsymbol{\alpha}$ können durch die Lösung der konvexen l_1-Norm-Optimierung geschätzt werden [11][16]:

$$\hat{\boldsymbol{\alpha}} = argmin. \|\mathbf{\Psi}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{2.9}$$

λ ist der Regularisierungsparameter und dient zur Anpassung der Gewichte zwischen der Rekonstruktionstreue $\|\mathbf{\Psi}\boldsymbol{\alpha} - x\|_2^2$ und dem Sparsity-Term $\|\boldsymbol{\alpha}\|_1$. Der geschätzte Vektor $\hat{\boldsymbol{\alpha}}$ enthält:

$$\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_{BW}|\hat{\boldsymbol{\alpha}}_{LF}|\hat{\boldsymbol{\alpha}}_{PLI}|\hat{\boldsymbol{\alpha}}_{HF}]^T \tag{2.10}$$

$\hat{\boldsymbol{\alpha}}_{BW}$, $\hat{\boldsymbol{\alpha}}_{LF}$, $\hat{\boldsymbol{\alpha}}_{PLI}$ beschreiben reellwertige Vektoren, welche zu den entsprecheneden Matritzen des Overcomplete Dictionaries korrespondieren. $\hat{\boldsymbol{\alpha}}_{HF}$ entspricht dem Residuum zur Repräsentation der zeitlichen Komponenten. Das EKG-Signal kann dargestellt werden durch:

$$\begin{aligned} \hat{x} \approx \mathbf{\Psi}\hat{\boldsymbol{\alpha}} &= [\mathbf{\Psi}_{BW}|\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]\hat{\alpha} \\ &= \mathbf{\Psi}_{BW}\hat{\boldsymbol{\alpha}}_{BW} + \mathbf{\Psi}_{LF}\hat{\boldsymbol{\alpha}}_{LF} + \mathbf{\Psi}_{PLI}\hat{\boldsymbol{\alpha}}_{PLI} + \mathbf{\Psi}_{HF}\hat{\boldsymbol{\alpha}}_{HF} \\ &= \hat{\mathbf{x}}_{BW} + \hat{\mathbf{x}}_{LF} + \hat{\mathbf{x}}_{PLI} + \hat{\mathbf{x}}_{HF} \end{aligned} \tag{2.11}$$

Wie die Gleichung 2.11 zeigt, kann das EKG-Signal als eine additive Zeitreihe repräsentiert werden. Damit ist gemeint, dass das EKG-Signal aus verschiedenen Komponenten besteht, die als Summe das EKG-Signal ergeben. $\hat{\mathbf{x}}_{BW}$ steht dabei für die Komponente des Signals, die das Baseline Wander darstellt. $\hat{\mathbf{x}}_{LF}$ beschreibt das Teilsignal, welches aus der T-Welle, P-Welle und niederfrequenten Bereichen des QRS-Komplexes besteht. $\hat{\mathbf{x}}_{PLI}$ definiert den Teil des Signals, welcher der Power-Line-Interferenz entspricht und $\hat{\mathbf{x}}_{HF}$ entspricht den spitzartigen, zeitlokalisierten Komponenten, wie hochfrequentem Rauschen oder den hochfrequenten Bereichen des QRS-Komplexes. Durch die Schätzung der Sparse-Koeffizienten ist es daher möglich, das Signal in einzelne Komponenten zu zerlegen und je nach vorliegendem Artefakt das EKG-Signal entsprechend ohne das Rauschen zu rekonstruieren. Auf die verschiedenen Vorgehensweisen wird im nächsten Unterkapitel näher eingegangen.

Entrauschungsprozedur

Je nach vorliegendem technischen Artefakt wird ein unterschiedliches Vorgehen gewählt. Die einzelnen Prozeduren werden in Tabelle 2.3 dargestellt. Es werden zwischen Situationen unterschieden, in denen entweder Baseline

Wander (BW) oder Power Line Interferenz (PLI) oder ein Muscle Artefact (MA) oder die Kombinationen aus den vorherigen vorliegt.

Liegt **nur** BW vor, so:

1) Bestimme das Dictionary $\mathbf{\Psi}_{BW}$.
2) Führe die Sparse Signal Decomposition auf $\mathbf{\Psi}_{BW}$ durch, um $\hat{\boldsymbol{\alpha}}_{BW}$ zu erhalten.
3) Schätze das Teilsignal $\hat{\mathbf{x}}_{BW}$.
4) Subtrahiere $\hat{\mathbf{x}}_{BW}$ vom EKG-Signal $\mathbf{x}$: $\tilde{\mathbf{x}} = x - \hat{\mathbf{x}}_{BW}$, sodass $\tilde{\mathbf{x}}$ das entrauschte Signal darstellt.

Liegt **nur** PLI vor, so:

1) Bestimme das Dictionary $\mathbf{\Psi}_{PLI}$.
2) Führe die Sparse Signal Decomposition auf $\mathbf{\Psi}_{PLI}$ durch, um $\hat{\boldsymbol{\alpha}}_{PLI}$ zu erhalten.
3) Schätze das Teilsignal $\hat{\mathbf{x}}_{PLI}$.
4) Subtrahiere $\hat{\mathbf{x}}_{PLI}$ vom EKG-Signal $\mathbf{x}$: $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}_{PLI}$, sodass $\tilde{\mathbf{x}}$ das entrauschte Signal darstellt.

Liegt BW+PLI vor, so:

1) Bestimme das kombinierte Dictionary $[\mathbf{\Psi}_{BW}|\mathbf{\Psi}_{PLI}]$.
2) Führe die Sparse Signal Decomposition auf $[\mathbf{\Psi}_{BW}|\mathbf{\Psi}_{PLI}]$ durch, um $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_{BW}|\hat{\boldsymbol{\alpha}}_{PLI}]$ zu erhalten.
3) Schätze die Teilsignale $\hat{\mathbf{x}}_{BW}$ und $\hat{\mathbf{x}}_{PLI}$.
4) Subtrahiere $\hat{\mathbf{x}}_{BW}$ und $\hat{\mathbf{x}}_{PLI}$ vom EKG-Signal $\mathbf{x}$: $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}_{BW} - \hat{\mathbf{x}}_{PLI}$, sodass $\tilde{\mathbf{x}}$ das entrauschte Signal darstellt.

Liegt MA oder MA+PLI vor, so:

1) Bestimme das Overcomplete Dictionary $[\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]$.
2) Führe die Sparse Signal Decomposition auf $[\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]$ durch, um $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_{LF}|\hat{\boldsymbol{\alpha}}_{PLI}|\hat{\boldsymbol{\alpha}}_{HF}]^T$ zu erhalten.
3) Schätze die Teilsignale $\hat{\mathbf{x}}_{LF}$, $\hat{\mathbf{x}}_{PLI}$ und $\hat{\mathbf{x}}_{HF}$.
4) Führe einen Algorithmus zur Erkennung des R-Peaks auf $\hat{\mathbf{x}}_{HF}$ durch.
5) Extrahiere aus $\hat{\mathbf{x}}_{HF}$ den QRS-Komplex mit einer definierten Breite von $L = 100ms$ über die gefundenen R-Peak-Instanzen $n_1, n_2, ..., n_R$.

$$QRS_{HF} := \begin{cases} \hat{\mathbf{x}}_{HF} & \text{, falls } n_i - \frac{L}{2} \leq \hat{\mathbf{x}}_{HF} \leq n_i + \frac{L}{2} \\ 0 & \text{, sonst.} \end{cases}$$

6) Addiere $\hat{\mathbf{x}}_{LF}$ und QRS_{HF}: $\tilde{\mathbf{x}} = \hat{\mathbf{x}}_{LF} + QRS_{HF}$, sodass $\tilde{\mathbf{x}}$ das entrauschte Signal darstellt.

Liegt BW+MA oder BW+MA+PLI vor, so:

1) Bestimme das Overcomplete Dictionary $[\mathbf{\Psi}_{BW}|[\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]$.
2) Führe die Sparse Signal Decomposition auf $[\mathbf{\Psi}_{BW}|\mathbf{\Psi}_{LF}|\mathbf{\Psi}_{PLI}|\mathbf{\Psi}_{HF}]$ durch, um $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_{BW}|\hat{\boldsymbol{\alpha}}_{LF}|\hat{\boldsymbol{\alpha}}_{PLI}|\hat{\boldsymbol{\alpha}}_{HF}]^T$ zu erhalten.
3) Schätze die Teilsignale $\hat{\mathbf{x}}_{BW}$, $\hat{\mathbf{x}}_{LF}$, $\hat{\mathbf{x}}_{PLI}$ und $\hat{\mathbf{x}}_{HF}$.
4) Führe einen Algorithmus zur Erkennung des R-Peaks auf $\hat{\mathbf{x}}_{HF}$ durch.
5) Extrahiere aus $\hat{\mathbf{x}}_{HF}$ den QRS-Komplex mit einer definierten Breite von $L = 100ms$ über die gefundenen R-Peak-Instanzen $n_1, n_2, ..., n_R$.

$$QRS_{HF} := \begin{cases} \hat{\mathbf{x}}_{HF} & \text{, falls } n_i - \frac{L}{2} \leq \hat{\mathbf{x}}_{HF} \leq n_i + \frac{L}{2} \\ 0 & \text{, sonst.} \end{cases}$$

6) Addiere $\hat{\mathbf{x}}_{LF}$ und QRS_{HF}: $\tilde{\mathbf{x}} = \hat{\mathbf{x}}_{LF} + QRS_{HF}$, sodass $\tilde{\mathbf{x}}$ das entrauschte Signal darstellt.

Tabelle 2.3: Entrauschungsprozeduren mithilfe der Sparse Signal Decomposition

2.3.2 *Wavelet-Transformation*

Die Wavelet-Transformation beschreibt eine effektive Methode zur Analyse nicht-stationärer Signale, wie es bei einer EKG-Zeitreihe der Fall ist [71]. Wavelets erweitern das Konzept der Fourier-Analyse auf allgemeinere orthogonale Basen und bieten einen Weg, der den Trade-off zwischen Frequenz-und Zeitauflösung angeht, welcher in Abbildung 2.7 dargestellt wird [8].

Die Abbildung kann folgendermaßen interpretiert werden: Die beiden Extreme werden durch die Zeitreihe a) und die Fourier-Transformation b) repräsentiert. Die Zeitreihe hat eine genaue Zeitauflösung, was bedeutet, dass jede Beobachtung einer genauen Zeit zugeordnet werden kann, jedoch keiner Frequenz. Die Fourieranalyse zeigt dagegen eine genaue Auflösung der Frequenzen. Allerdings kann keine Aussage darüber gemacht werden, zu welchem Zeitpunkt die Frequenzen vorliegen. Das Spektogramm (c) berücksichtigt sowohl die Zeit- als auch die Frequenzinformation, jedoch mit geringerer Auflösung in beiden Bereichen. Ein anderer Weg den Trade-off zu überwinden wird durch die Wavelets (d) ermöglicht. Für niedrige Frequenzen liegt zwar eine genau Frequenzauflösung vor, allerdings auch eine geringe Zeitauflösung, da niedrige Frequenzen sich nicht schnell ändern, bezogen auf die Zeitkomponente. Je höher die Frequenz wird, desto genauer wird die Zeitauflösung, jedoch ungenauer die Frequenzauflösung. Dies ist vorteilhaft, da sich höhere Frequenzen schneller bezüglich der Zeitinformation ändern, wodurch ein genauerer Zeitbezug notwendig wird.

Durch die Möglichkeit sowohl die Frequenz- als auch die Zeitinformation einer Zeitreihe in unterschiedlicher Genauigkeit darzustellen, ist die Wavelet-Transformation ebenfalls eine Methode zur Elimination von techni-

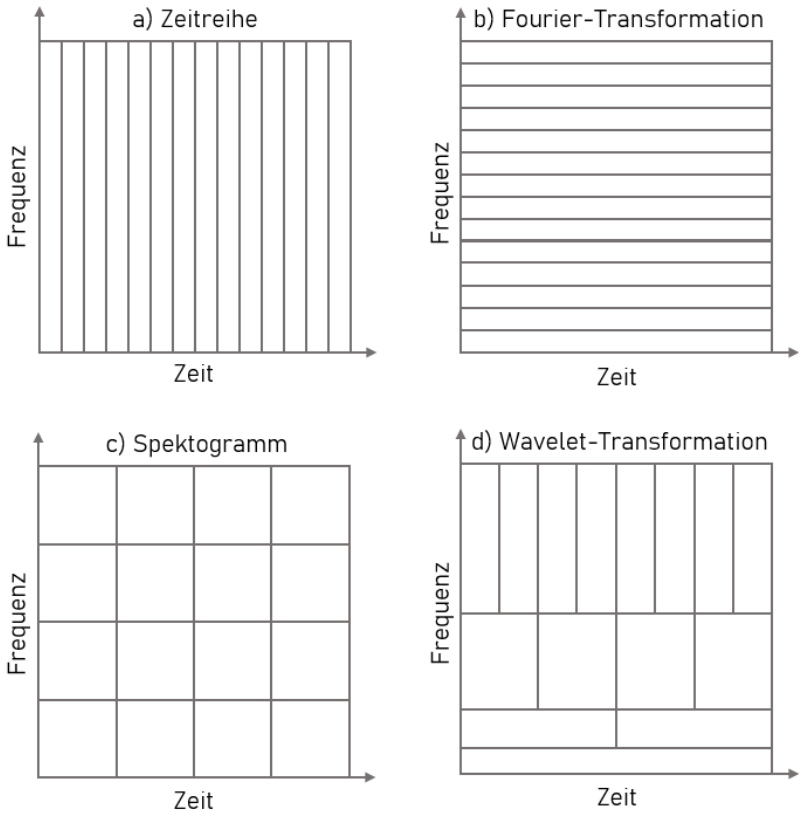

Abbildung 2.7: Illustration der Limitationen und Unsicherheiten in der Zeit-Frequenz-Analyse
Quelle: In Anlehnung an Brunten und Kutz, 2019 [8]

schen Artefakten, da Vorwissen zu bestimmten Frequenzeigenschaften der Artefakte besteht. Das bedeutet, dass über die Wavelet-Analyse sich die einzelnen Frequenzbereiche lokalisieren und eliminieren lassen. Die Methode wird ähnlich zum vorherigen Kapitel in einzelne Unterkapitel gegliedert. Zunächst wird die Zieldarstellung der sparsen Repräsentation beschrieben, damit eine Entrauschung ermöglicht wird. Die Zieldarstellung entspricht der Repräsentation als Wavelet-Transformierte. Daraufhin wird ein Wavelet, welches zur Erreichung der Zieldarstellung eingesetzt wird, beschrieben. Dabei wird erläutert, wann eine Funktion als Wavelet fungieren kann und Beispiele für die Gestalt eines Wavelets gegeben. Anschließend wird auf die Wavelet-Transformation eingegangen und zum Schluss die darauf basierende Entrauschungsprozedur über eine Filterbank beschrieben.

Sparse Repräsentation

Bei der Wavelet-Transformation wird dem Ansatz gefolgt, die Zeitreihe $x = (x_1, x_2, ..., x_P)^T$, welche im Folgenden als eine Funktion $f(t) \in L_2(\mathbb{R})$ aufgefasst wird, durch Koeffizienten zu repräsentieren, die die Orthogonalität zwischen $f(t)$ und verschiedenen skalierten Versionen eines Wavelets $\Psi_{j,k}$

beschreiben [46]. t entspricht der zeitlichen Komponente. Ziel ist die Darstellung der Zeitreihe somit als:

$$W_\Psi(f)(j,k) = \langle f, \Psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t)\bar{\Psi}_{j,k}(t)dt \tag{2.12}$$

$W_\Psi(f)(j,k)$ beschreibt die Wavelet-Transformierte des Signals $f(t)$ und gibt die zur Repräsentation genutzten Koeffizienten an. Da die Koeffizienten die Orthogonalität zwischen $f(t)$ und $\Psi_{j,k}$ beschreiben, handelt es sich bei den Koeffizienten-Vektoren $W_\Psi(f)(j,k)$ um eine sparse Repräsentation. Die meisten Koeffizienten liegen bei oder nahe Null.

Design des Wavelets

Der essentielle Schritt der Wavelet-Transformation ist die richtige Wahl von Ψ. Es gibt verschiedene Wavelets, die unterschiedliche Eigenschaften aufweisen und dadurch für unterschiedliche Anwendungsgebiete geeignet sind [46]. Die grundlegende Idee der Wavelet-Analyse liegt darin, mit einer quadrat-integrierbaren Funktion $\Psi(t)$ zu starten und basierend darauf eine Familie von skalierten und übersetzten Versionen der Funktion zu generieren [8]. Dadurch basieren alle orthogonalen Basisfunktionen innerhalb der Wavelet-Analyse auf einem sogenannten Mother-Wavelet $\Psi(t)$:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right) \tag{2.13}$$

Durch den Parameter a wird die Funktion skaliert. Der Parameter b dient der Übersetzung [8]. Beispielsweise könnten a und b so gewählt werden, dass $\Psi_{a,b}(t)$ zu jedem Segment in d) aus Abbildung 2.7 passt.

Es gibt zwei Bedingungen, die eine Funktion erfüllen muss, damit diese als Wavelet fungieren kann. Zum einen muss sie die Zulässigkeitsbedingung 2.14 erfüllen [22]:

$$\int \frac{|\Psi(\omega)|^2}{|\omega|}d\omega < +\infty \tag{2.14}$$

$\Psi(\omega)$ steht für die Fourier-Transformation von $\Psi(t)$. Durch die Bedingung 2.14 wird impliziert, dass die Fourier-Transformation von $\Psi(t)$ bei einer Frequenz von Null verschwindet [22], d.h.

$$|\Psi(\omega)|^2_{\omega=0} = 0 \tag{2.15}$$

Die Gleichung 2.15 führt zu der Eigenschaft eines Mother-Wavelets, dass der Mittelwert des Wavelets Null entspricht [22]:

$$\int \Psi(t)dt = 0 \tag{2.16}$$

Die zweite Bedingung eines Wavelets ist die Regelmäßigkeitsbedingung[3]. Diese besagt, dass ein Wavelet sowohl im Zeit- als auch im Frequenzbereich eine gewisse Glätte und Konzentration aufweisen sollte [35].

3 Weiterführende Informationen zur Regelmäßigkeitsbedingung sind [35] zu entnehmen.

Das einfachste Beispiel eines Wavelets ist das Haar-Wavelet, welches folgendermaßen definiert wird:

$$\Psi(t) := \begin{cases} 1 & \text{, falls } 0 \leq t < \frac{1}{2} \\ -1 & \text{, falls } \frac{1}{2} \leq t < 1 \\ 0 & \text{, sonst.} \end{cases} \tag{2.17}$$

Die Abbildung 2.8 zeigt die drei Haar-Wavelets $\Psi_{1,0}$, $\Psi_{\frac{1}{2},0}$ und $\Psi_{\frac{1}{2},\frac{1}{2}}$. Sie repräsentieren die zwei unteren Schichten aus d) der Abbildung 2.7. Für jede höhere Frequenzschicht halbiert sich das Haar-Wavelet, sodass eine doppelte Zeitgenauigkeit, aber eine halbierte Frequenzgenauigkeit resultiert [8]. Alle am Ende entstehenden Wavelets sind orthogonal zueinander und bilden eine hierarchische Basis des Signals [8]. Insbesondere die Orthogonalität ist essentiell für die im nächsten Unterkapitel beschriebene DWT [8].

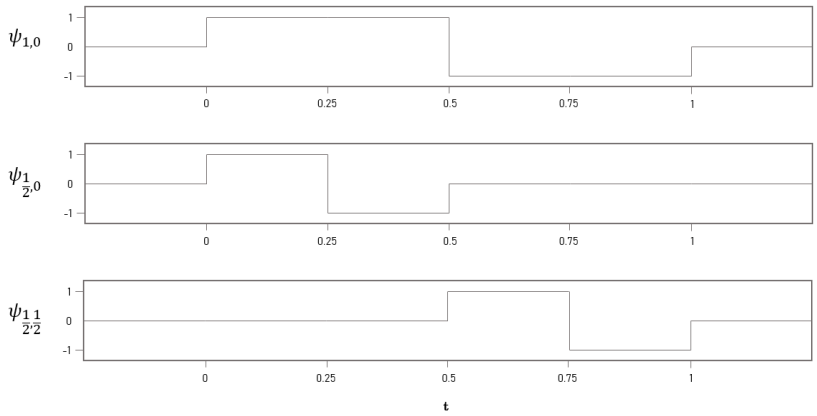

Abbildung 2.8: Die drei Haar-Wavelets der ersten zwei Level der Wavelet-Transformation (d) aus Abbildung 2.7
Quelle: In Anlehnung an Brunton und Kutz, 2019 [8]

Neben dem Haar-Wavelet gibt es noch weitere Wavelet-Familien, die für unterschiedliche Anwendungsfälle genutzt werden. Sie unterscheiden sich alle in ihrer Form, Glätte und Kompaktheit [22]. Alle erfüllen die zuvor beschriebenen Wavelet-Eigenschaften.

Beispiele für verschiedene Mother-Wavelets sind der Abbildung 2.9 zu entnehmen. Die am häufigsten gewählten Mother-Wavelet-Basisfunktionen zur Entrauschung von EKG-Daten sind die Daubechies-Filter, Symmlet-Filter, Coiflet-Filter, Battle-Lamerie-Filter, Beylkin-Filter, Vaidyanathan-Filter und die Discrete Meyer-Filter [37, 50, 66].

Innerhalb einer Wavelet-Familie kann es verschiedene Wavelet-Unterkategorien geben, welche sich anhand der Koeffizienten-Anzahl (vanishing moments) und des Decomposition-Levels unterscheiden [37].

Kontinuierliche und Diskrete Wavelet-Transformation

Bei der Wavelet-Transformation wird zwischen der Kontinuierlichen und der Diskreten Wavelet-Transformation unterschieden. Die Kontinuierliche

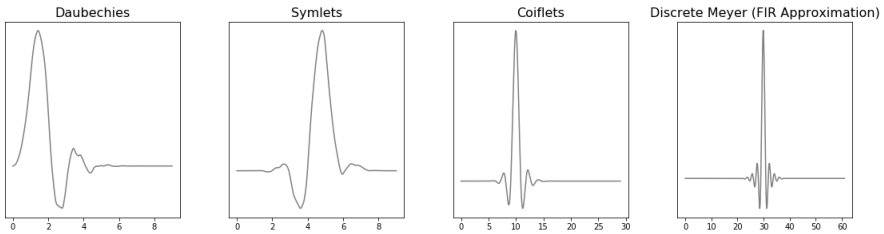

Abbildung 2.9: Beispiele für vier verschiedene Wavelet-Familien, welche zur diskreten Wavelet-Transformation genutzt werden können
Quelle: Eigene Darstellung mithilfe des Python-Pakets PyWavelets [54]

Wavelet-Transformation (CWT) bildet dabei die Grundlage für die DWT. Die CWT ist gegeben durch [8]:

$$W_\Psi(f)(a,b) = \langle f, \Psi_{a,b} \rangle = \int_\infty^\infty f(t)\bar{\Psi}_{a,b}(t)dt \tag{2.18}$$

$\bar{\Psi}_{a,b}$ entspricht der komplexen Konjugation von $\Psi_{a,b}$. Die Inverse Kontinuierliche Wavelet-Transformation ist geben durch:

$$f(t) = \frac{1}{C_\Psi} \int_{-\infty}^\infty \int_{-\infty}^\infty W_\Psi(f)(a,b)\Psi_{a,b}(t)\frac{1}{a^2}da\,db \tag{2.19}$$

Die Beschreibung „diskret“ tätigt bei der DWT eine andere Aussage als beispielsweise bei der Fourier-Transformation oder ihren verwandten Spaktral-Transformationen. Die DFT beschreibt die Anwendung der Fourier-Transformation auf diskrete Signale zum Erhalt eines diskreten Frequenzspektrums. Die DWT wird dagegen auf kontinuierliche Signale angewendet und erzeugt dabei eine invertierbare Signaldarstellung für die diskreten Parameterwerte zur Skalierung des Wavelets. Das bedeutet, dass die DWT der für diskrete Transformationsvariablen berechneten CWT entspricht. Die Grundlage dafür bildet die notwendige Orthogonalitätseigenschaft der zugrunde liegenden Basisfunktion, da dadurch das Signal durch die sogenannten DWT-Koeffizienten dargestellt werden kann und es eine verlustfreie Signalrekonstruktion ermöglicht [4].

Die DWT ist gegeben durch [8]:

$$W_\Psi(f)(j,k) = \langle f, \Psi_{j,k} \rangle = \int_{-\infty}^\infty f(t)\bar{\Psi}_{j,k}(t)dt \tag{2.20}$$

$\Psi_{j,k}(t)$ beschreibt dabei ein Wavelet der diskreten Wavelet-Familie [8]:

$$\Psi_{j,k}(t) = \frac{1}{a^j}\Psi\left(\frac{t-kb}{a^j}\right) \tag{2.21}$$

Zur Rekonstrunktion der Zeitreihe aus den Koeffizienten $W_\Psi(f)(j,k)$ wird die Rechenvorschrift der Inversen Diskreten Wavelet-Transformation herangezogen, welche definiert ist als:

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} W_\Psi(f)(j,k)\Psi_{j,k}(t) \tag{2.22}$$

Die DWT wird oftmals zur Verringerung der Rechenkomplexität durch die Diskretisierung der Transformationsvariablen eingesetzt.

Entrauschungsprozedur

Zur Entrauschung von EKG-Signalen kann die DWT als eine Filterbank implementiert werden. Unter einer Filterbank wird eine Anordnung aus Tief-, Band- und Hochpass-Filtern verstanden, mit welchen Signale spektral zerlegt oder aus den Spektralanteilen wieder zusammengesetzt werden können [45]. Ein Tiefpassfilter beschreibt einen Filter, welcher nur die tiefen Frequenzen eines Signals erfasst. Ein Hochpassfilter dagegen erfasst lediglich die hochen Frequenzen eines Signals. Zur Übertragung der DWT auf eine Filterbank entwickelte S. Mallat ein Verfahren zur iterativen Berechnung der Wavelet-Koeffizienten, welches zur *Multiresolution Analysis* eingesetzt wird. Mithilfe einer Skalierungsfunktion, auch *Father-Wavelet* genannt, ist es möglich ein Signal in verschiedenen Auflösungen darzustellen, sodass die DWT als Tiefpass- und Hochpass-Filter implementiert werden kann. Die Ergebnisse dieser Filter entsprechen den DWT-Koeffizienten. Die Möglichkeit, mithilfe der DWT und einer Skalierungsfunktion Hoch- und Tiefpassfilter zu konstruieren, wird als gegeben angenommen.

Bei der Implementierung zur Entrauschung von EKG-Daten wird auf ein Filterbank-Paar zurückgegriffen, wobei die eine Filterbank für die schrittweise Zerlegung und die andere Filterbank für die schrittweise Rekonstruktion, oder auch Synthese genannt, verantwortlich ist [64]. In jedem Zerlegungsschritt der ersten Filterbank werden die zwei Analysefilter - Tiefpass- und Hochpassfilter - auf das Signal angewendet, sodass das Signal in zwei Subsignale unterteilt wird. Jedes der Subsignale beinhaltet die Hälfte der Frequenzen. Das eine SUbsignal die hohen, das andere Subsignal die niedrigen Frequenzen. Die Ergebnisse der Filter werden mit fester Rate heruntergetastet und die Analysefilter auf das Ergebnis des Tiefpassfilters wiederholt angewendet, um so eine definierte Menge von Teilbandsignalen zu erzeugen [64]. Das Verfahren wird auch als *Subband Coding* bezeichnet. Das Abwärtsabtasten entsprechend einer Abtrastrate dient der Verringerung oder Entfernung der in den Teilbandsignalen enthaltener Redundanzen, da nur unterabgetastete Teilbandsignale wieder fehlerfrei rekonstruiert werden können [45]. Praktisch bedeutet dies, dass jeder x-te Punkt (entsprechend der Abtastrate) aus dem Signal vor der Weiterverarbeitung entfernt wird. Dieses Vorgehen wird *Downsampling* genannt. Jedes der Teilsignalbänder besitzt Informationen zu einem bestimmten Frequenzbereich des Eingangssignals [45]. Zur perfekten Rekonstruktion wird eine zweite Filterbank implementiert, die

aus derselben Anzahl an Systhesefiltern besteht, wie zuvor Analysefilter eingesetzt wurden [64]. Die einzelnen Teilbandsignale werden daraufhin mit derselben Rate hochgetastet (*Upsampling*) und auf einen Synthesefilter angewendet. Eine schematische Darstellung einer Filterbank-Implementation zur Entrauschung von EKG-Daten wird durch Abbildung 2.10 dargestellt, wobei die Filterbank zur Zerlegung als Dekomposition und die Filterbank zur Wiederherstellung des Signals als Rekonstrunktion beschrieben wird. Innerhalb der Abbildung wird eine Abtastrate von 2 gewählt.

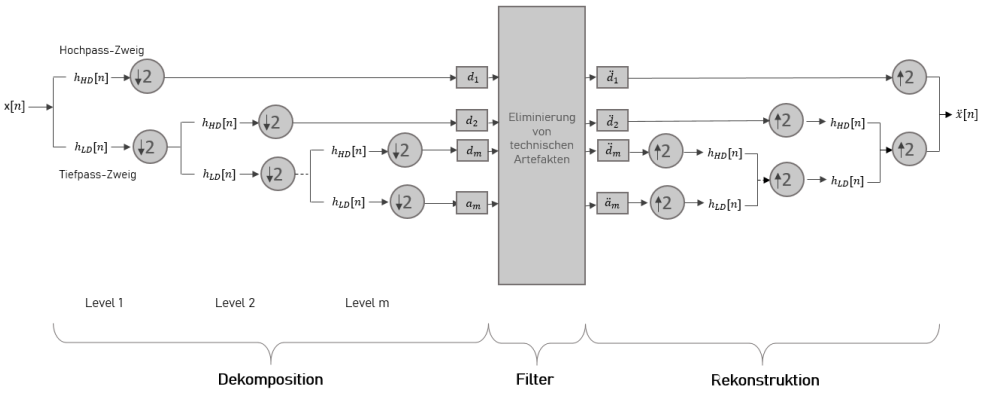

Abbildung 2.10: Filterbank-Schema für eine Down- und Upsampling-Rate von 2
Quelle: In Anlehung an Alessio, 2016 [4] und Nguyen, 2020 [50]

Die in Abbildung 2.10 gezeigten Analyseergebnisse $\mathbf{d_1}$ bis $\mathbf{d_m}$ und $\mathbf{a_m}$ entsprechen den Detailkoeffizienten (**d**), bzw. den Approximationskoeffizienten (**a**), welche bei der Wavelet-Transformation entstehen. Da in jedem Zerlegungsschritt, auf den ein weiterer Zerlegungsschritt folgt, die Ergebnisse des Tiefpassfilters weiterverarbeitet werden, resultieren hier lediglich die Ergebnisse des Hochpassfilters in Form der Wavelet-Koeffizienten. Diese werden Detailkoeffizienten genannt. Im letzten Zerlegungsschritt resultieren sowohl die Ergebnisse des Hochpass- als auch des Tiefpassfilters, sodass in diesem Dekompositionlevel sowohl ein Detailkoeffizienten-Vektor, als auch als Ergebnis des Tiefpassfilters ein Approximationskoeffizienten-Vektor in Form von Wavelet-Koeffizienten entstehen. Diese können den Frequenzbändern aus Tabelle 2.4 zugeordnet werden.

Ein Signal mit der Sampling-Rate von 512 Werten pro Sekunde besitzt ein entsprechendes Frequenzband von -256 bis 256 Hz. Der erste Hochpass-Filter ergibt eine Frequenz von 128 bis 256 Hz. Der erste Tiefpassfilter von 0 bis 128 Hz. Das Ergebnis des Tiefpassfilters wird wieder auf einen Hoch- und Tiefpassfilter angewandt, wodurch sich wiederum ein halbiertes Frequenzband ergibt. Dies wird so lange wiederholt, bis das maximale Dekompositionslevel erreicht ist. Das daraus resultierende Ergebnis des Tiefpassfilters entspricht dem Frequenzband des Approximationskoeffizienten-Vektors. Durch das Auslassen jedes zweiten Abtastwertes halbiert sich mit jedem Dekompositionslevel die Zeitauflösung, jedoch verdoppelt sich gleichzeitig die Frequenzauflösung, da sich die zu analysierende Bandbreite halbiert.

Level	DWT-Koeffizienten	Frequenzbereich (in Hz)
1	$\mathbf{d_1}$	128-256
2	$\mathbf{d_2}$	65-128
3	$\mathbf{d_3}$	32,5-65
4	$\mathbf{d_4}$	16,25-32,5
5	$\mathbf{d_5}$	8,125-16,25
6	$\mathbf{d_6}$	4,063-8,125
7	$\mathbf{d_7}$	2,031-4,063
8	$\mathbf{d_8}$	1,016-2,031
8	$\mathbf{a_8}$	0-1,016

Tabelle 2.4: Frequenzbereiche der DWT-Koeffizienten der acht Dekompositionslevel bei einem 512 Hz Signal
Quelle: In Anlehnung an Nguyen, 2020 [50])

Da Vorwissen zu den Frequenzeigenschaften bestimmter Artefakten besteht, können durch die einzelnen Koeffizienten diese lokalisiert und eliminiert werden. Basline-Wander bewegt sich im Bereich von 0 bis 1 Hz und PLI zwischen 47 und 53 Hz, wodurch der Detailkoeffizienten-Vektor d_3 und der Approximationskoeffizienten-Vektor a_8 vor der Rekonstruktion als Null-Vektoren gesetzt werden, was wiederum zur Elimination der beiden Artefakte führt.

2.3.3 *Empirical Mode Decomposition*

Die Empirical Mode Decomposition (EMD) dient der Analyse von komplexen, nicht-stationären Zeitreihen durch die Zerlegung in eine Gruppe lokal orthogonaler Basisfunktionen, sogenannte Intrinsic Mode Functions (IMF) (IMFs). Durch die Orthogonalitätseigenschaft wird eine verlustfreie Rekonstruktion ermöglicht [75]. Die vollständig datengesteuerte und unüberwachte Signalzerlegung führt dazu, dass kein a priori definiertes Basissystem benötigt wird [75]. Die Eigenschaften der EMD erinnern daher an die Wavelet-Zerlegung, die Interpretation der resultierenden Funktionen ist jedoch eine andere [75].

Zur Beschreibung der Elimination von technischen Artefakten über die EMD wird zunächst die Zieldarstellung der Zeitreihe, wie auch die Definition einer IMF erläutert. Daraufhin werden die Schritte des EMD-Algorithmus beschrieben und im Nachhinein zwei aufeinander aufbauende Erweiterungen der EMD vorgestellt, da sie vorzugsweise zur Elimination der technischen Artefakte eingesetzt werden. Zum Schluss wird auf die Entrauschung von EKG-Daten über die EMD eingegangen.

Repräsentation

Ziel der EMD ist die Darstellung der Zeitreihe x als Summe der IMFs und dem finalen Residuum:

$$x = \sum_{i=1}^{N} IMF_i + r_N \tag{2.23}$$

r_N repräsentiert dabei das Residuum. N gibt die Anzahl an IMFs an. Die Amplituden und Frequenzen der IMFs können sich über die Zeitkomponente hinweg ändern [75].

Die IMFs haben zwei grundlegende Eigenschaften [73, 75]:

1. Eine IMF besitzt nur ein Extremum zwischen zwei nebeneinander liegenden Nullstellen, d.h. die Anzahl an Minima und die Anzahl an Maxima unterscheiden sich höchstens um Eins.

2. Der Mittelwert jeder IMF ist gleich Null.

Der EMD-Algorithmus basiert auf der Annahme, dass jede nicht-stationäre und nicht-lineare Zeitreihe aus verschiedenen einfachen intrinsischen Schwingungsmodi besteht [75]. Die IMFs werden über einen *Shifting-Prozess* identifiziert, indem sogenannte Reitwellen, d.h. Schwingungen ohne eine Nullstelle zwischen zwei Extrema, eliminiert werden [75]. Der EMD-Algorithmus berücksichtigt somit Signalschwingungen auf einer sehr lokalen Ebene und trennt die Daten in lokale, nicht-überlappende Zeitskalenkomponenten [75]. Der Shifting-Prozess zur Identifizierung der IMFs wird im nächsten Unterkaitel erläutert.

Empirical Mode Decomposition

Der Algorithmus ist in einzelne Schritte zu unterteilen, die im Folgenden dargestellt werden [73, 75]:

1. Finden aller Extrempunkte (Minima und Maxima) in der Zeitreihe $x(t)$.

2. Kubische Spline-Interpolation über
   - **A:** Alle Maxima, zum Erhalt der oberen Hülle $\mu_0(t)$
   - **B:** Alle Minima, zum Erhalt der unteren Hülle $d_0(t)$

3. Mittelwertbildung aus den beiden gefundenen Hüllen:

$$m_o(t) = \frac{1}{2}\left(\mu_0(t) + d_0(t)\right) \tag{2.24}$$

4. Subtraktion der Werte m_o von $m_o(t)$, die den Zeitpunkten der Zeitreihe $x(t)$ entsprechen:

$$h_1 = x - m_o \tag{2.25}$$

5. Erfüllt h_1 die IMF-Kriterien, so setze $h_1 = IMF_1$. Andernfalls wird $h_1(t)$ als Input des Shifting-Prozesses betrachtet und die Schritte 1 bis 4 wiederholt. Es entsteht eine neue Funktion $h_{11}(t)$. Dieser Prozesss wird so lange wiederholt, bis entweder h_{1k} den IMF-Kriterien entspricht, oder eine definierte Abbruchbedingung (im Allgemeinen Standardabweichungskriterien) erfüllt ist.

6. Subtraktion der IMF_1 von $x(t)$, um das Residuum $r_1(t)$ zu erhalten:

$$r_1 = x - IMF_1 \tag{2.26}$$

7. Nutzen des Residuums r_1 als neuen Input. Wiederholung der Schritte 1 bis 6, um die $IMF_2, IMF_3, ..., IMF_N$ zu erhalten, bis r_N eine Konstante, eine Funktion mit monotoner Steigung, oder eine Funktion mit nur einem Extremum wird.

Nach der Zerlegung in einzelne IMFs kann die Zeitreihe x dargestellt werden als

$$x = \sum_{j=1}^{N} IMF_j + r_N(t), \tag{2.27}$$

wobei N die Anzahl an IMFs repräsentiert und r_N das finale Residuum darstellt.

Der Hauptnachteil des EMD-Algorithmus ist der sogenannte *Mode-Mixing-Effekt*. Dieser besagt, dass Schwingungen unterschiedlicher Zeitskalen innerhalb einer IMF koexistieren können oder eine Schwingung mit derselben Zeitskala verschiedenen IMFs zugewiesen werden kann [73]. Im nächsten Kapitel wird eine Lösung für diese Problematik erläutert.

Ensemble Empirical Mode Decomposition

Zur Umgehung des Mode-Mixing-Effekts entwickelten Wu und Huang [72] einen Ensemble-Empirical-Mode-Decomposition-Algorithmus, welcher die EMD auf einem Set von verrauschten Kopien des Originalsignals durchführt und die einzelnen IMFs durch die Mittelwertbildung erhält. Durch das Hinzufügen von weißem gaußschen Rauschen wird der gesamte Zeit-Frequenz-Raum gefüllt, was die Wahrscheinlichkeit von regelmäßigeren Modi mit ähnlichen Skalen über die gesamte Zeitspanne erhöht [73].

Die genaue Abfolge der Schritte wird im Folgenden erläutert [73]:

1. I-mal: Hinzufügen von weißem gaußschem Rauschen $n_i(t)$, mit einem Mittelwert von Null und einheitlicher Varianz, zum Originalsignal x:

$$x_i = x + n_i(t) \tag{2.28}$$

2. Zerlegung jeder Zeitreihe x_i mit $i = 1, ..., I$ über den EMD-Algorithmus in die einzelnen IMF_{ij}, wobei $j = 1, ..., N$ der IMF-Nummer entspricht.

3. Berechnung IMFs als Mittelwert aus dem Ensemble:

$$\overline{IMF_j} = \frac{1}{I}\sum_{i=1}^{I} IMF_{ij} \tag{2.29}$$

Durch das hinzugefügte weiße gaußsche Rauschen kann zwar der Mode-Mixing-Effekt umgangen werden, jedoch kann dieses Rauschen durch eine endliche Anzahl an Ensemble-Komponenten nicht vollständig eliminiert werden, sodass ein Rekonstruktionsfehler entsteht [73]. Je höher die Anzahl an Ensemble-Komponenten, desto kleiner wird der Rekonstruktionsfehler. Die Vergrößerung des Ensembles führt jedoch auch proportional zu einer Erhöhung der Rechenkosten [73]. Zusätzlich ist nicht gewährleistet, dass bei jedem neuen EMD-Durchlauf dieselbe Anzahl an IMFs resultiert.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

Die Problematiken der Erhöhung der Rechenkosten und der Ungewissheit, ob im nachfolgenden EMD-Durchlauf dieselbe IMF-Anzahl entsteht, wurden durch Torres et al. [69] erkannt und der Zerlegungsprozess durch die entwickelte Erweiterung Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) optimiert. Der Mode-Mixing-Effekt kann durch diesen Algorithmus ebenfalls überwunden werden, der Rechenaufwand wird stark reduziert und der Rekonstruktionsfehler wird nahezu Null [73].

x entspricht der Zeitreihe, die in IMFs zerlegt werden soll. ϵ_j entspricht der Amplitude des hinzugefügten weißen gaußschen Rauschens $n_i(t)$. Die einzelnen Realisationen werden definiert als $x_i = x + \epsilon_0 \cdot n_i(t)$, wobei $n_i(t)$ einen Mittelwert von Null und eine Einheitsvarianz aufweist. ϵ_0 wird bei Torres et al. als $0,02$ festgelegt [69]. Python-Implementationen nutzen einen Default-Wert von $0,005$ [53].

Die CEEMDAN wird in folgende Schritte unterteilt [29, 73]:

1. Berechnung der ersten IMF als Mittelwert der ersten Modi der Realisationen von x_i mit $i = 1, ..., I$:

$$\widetilde{IMF_1} = \frac{1}{I}\sum_{i=1}^{I} IMF_{i1} \tag{2.30}$$

2. Berechnung des ersten Residuums:

$$r_1 = x - \widetilde{IMF_1} \tag{2.31}$$

3. Berechnung der zweiten IMF über den ersten Modus von $r_1 + \epsilon_1 IMF_1(n_i(t))$ mittels des EMD, sodass die zweite CEEMDAN-IMF definiert wird als:

$$\widetilde{IMF_2} = \frac{1}{I}\sum_{i=1}^{I} IMF_1(r_1 + \epsilon_1 IMF_1(n_i(t))) \tag{2.32}$$

Das weiße gaußsche Rauschen wird, wie in der Formel 2.32 deutlich wird, ebenfalls über den EMD-Algorithmus zerlegt, sodass $n_i \approx IMF_1(n_i) + IMF_2(n_i) + ... + IMF_J(n_i)$ gilt.

4. Für $j = 2, ..., J$, ist das j-te Residuum gegeben durch:

$$r_j = r_{j-1} - \widetilde{IMF_j} \tag{2.33}$$

5. Nutzen der EMD zur Berechnung der j-ten IMF der Realisationen $r_j + \epsilon_j IMF_j(n_i(t))$, mit $i = 1...I$:

$$\widetilde{IMF_{j+1}} = \frac{1}{I} \sum_{i=1}^{I} IMF_1(r_j + \epsilon_j IMF_j(n_i(t))), \tag{2.34}$$

wobei $IMF_j(\cdot)$ dem j-ten Modus erhalten über den EMD-Algorithmus entspricht.

6. Wiederholung der Schritte 4 und 5 solange, bis r_N eine Zeitreihe mit weniger als drei lokalen Extrema darstellt oder ein Stoppkriterium erfüllt wird. Das finale Residuum wird definiert durch:

$$r_N = x - \sum_{j=1}^{I} \widetilde{IMF_j} \tag{2.35}$$

N beschreibt die Anzahl an resultierenden IMFs. Die Zeitreihe kann rekonstruiert werden durch:

$$x = \sum_{j=1}^{N} \widetilde{IMF_j} + r_N(t) \tag{2.36}$$

Entrauschungsprozedur

Nach der Zerlegung einer Zeitreihe in die einzelnen IMFs über einen der vorgestellten Algorithmen, kann die Entrauschung der Zeitreihen erfolgen. Zur Elimination werden zum einen verrauschte IMFs identifiziert und von der Rekonstruktion ausgeschlossen. Dies wird insbesondere zur Entfernung des Baseline Wanders durchgeführt. Zum anderen werden mittels eines weiteren Algorithmus verrauschte IMFs bearbeitet, wie beispielsweise mittels des Wavelet-Thresholding-Ansatzes [73, 75]. Zum Schluss wird das EKG-Signal durch die Summe der übergebliebenen IMFs rekonstruiert, sodass ein entrauschtes Signal dargestellt werden kann.

2.4 HERZSCHLAGERKENNUNG UND -SEGMENTIERUNG

Die Herzschlagerkennung und -segmentierung entspricht der Erkennung und der darauf folgenden Extrahierung eines Herzschlages, welcher in der Klassifikation genutzt werden kann [37]. Diese Erkennung kann in drei Teile

untergliedert werden: Die Erkennung der P-Welle, die Erkennung des QRS-Komplexes und die Erkennung der T-Welle. Standardmäßig ist der Startpunkt eines Herzschlagsegments als Beginn der P-Welle und der Schlusspunkt als Ende der T-Welle zu verzeichnen [37]. Im Vergleich zur P- und T-Welle ist der QRS-Komplex - genau genommen der R-Peak - algorithmisch gut zu erfassen. Einige Studien zu R-Peak-Erkennungsalgorithmen sind in der Literatur zu finden und werden im Folgenden erläutert..

R-Peak-Erkennungs-Algorithmen

Der Pan-Tompkins-Algorithmus [51] ist einer der populärsten und ältesten Algorithmen, der zur R-Peak-Erkennung implementiert wurde. Aufgrund der Robustheit und Recheneffizienz wird er immer noch im Original oder in einer modifizierten Version [1, 13, 14] in vielen Anwendungen eingesetzt. Die Hauptkomponente des Algorithmus bildet eine Filterbank, welche aus Bandpassfiltern, einem Differenzierer, einem Quadrierungsfilter und einem Moving-Average-Integrator besteht, um das Signal so zu reduzieren, dass nur noch die R-Wellen-Information zurückbleibt [37]. Algorithmen, die an den Pan-Tompkins-Algorithmus angelehnt sind, nutzen zudem einen Amplitudenschwellenwert und einen RR-Intervalllängen-Schwellenwert, um die falsch positiven Ergebnisse zu reduzieren [13, 14]. Das RR-Intervall steht für den Abstand zweier aufeinanderfolgende R-Peaks.

Elgendi et al. [18] nutzen zur R-Peak-Erkennung einen zweifachen Moving-Average-Filter. Das Fenster des einen Filters entspricht der Breite eines Herzschlags, der zweite der Breite des QRS-Komplexes. Mithilfe der beiden Filter können nun die Blöcke erfasst werden, bei deren der Filter mit der Breite des QRS-Komplexes größer ist als der Filter mit der Breite des Herzschlags. Durch einen *Threshold* werden anschließend die Stellen als R-Peak lokalisiert, bei denen der Block eine kleinere Breite als 44 Datenpunkte aufweist.

Ein weiterer Ansatz zur Erkennung des QRS-Komplexes (bspw. Zidelmanl et al. [76] und Bouaziz et al. [47]) basiert auf der DWT in Verbindung mit einem Threshold.

Manikandan et al. [41] entwickelten einen Algorithmus mit dem Namen *Shannon energy envelop and Hilbert-transform (SEEHT)*. Hauptkomponenten des Algorithmus sind ein Shannen-Energy-Envelop-Schätzer, die Hilbert-Transformation und ein Moving-Average-Filter.

In der Literatur lassen sich zahlreiche weitere Algorithmen zur R-Peak-Erkennung finden, auf die jedoch aus Umfangsgründen im Rahmen dieser Arbeit nicht näher eingegangen wird.

Herzschlagsegmentierung

Wie oben bereits erläutert, basiert die Segmentierung auf der P- und T-Welle, da sie Start- und Stoppunkt des Herzschlags angeben. Zur Erkennung der P- und T-Welle wurden ebenfalls Verfahren entwickelt, die auf der Wavelet-Transformation basieren. Jedoch erzielen diese Verfahren lediglich auf normalen Herzschlägen zuverlässige Ergebnisse. Bei Anomalien können die P-

und T-Welle daher nicht gut automatisiert gefunden werden, weshalb eine Extraktion über diesen Weg große Schwierigkeiten bereitet. Aus diesem Grund wird zur Segmentierung oftmals auf eine manuelle Annotation durch Experten [74] oder ein festes Fenster [68, 74] zurückgegriffen. Ein festes Fenster bedeutet, dass auf Basis des erkannten R-Peaks eine definierte Breite links des Peaks und eine definierte Breite rechts davon als ein Herzschlag-Segment extrahiert wird. Die Algorithmen zur R-Peak-Erkennung liefern auch bei abnormalen Herzschlägen zuverlässige Ergebnisse.

2.5 METHODEN ZUR FEATURE-EXTRAKTION

Der Schritt der Feature-Extraktion beschreibt einen Prozess der Dimensionsreduktion, bei dem ein Satz von Rohdaten zur Verarbeitung auf eine besser verwaltbare Anzahl an Feature reduziert wird. Die Feature sollen die für das angestrebte Ziel entsprechend relevantesten Informationen der Daten möglichst gut repräsentieren. Dabei wird der Schritt in zwei Unterschritte aufgeteilt [25]: *Feature-Konstruktion* und *Feature-Selektion*.

Im Schritt der Feature-Konstruktion geht es darum, entweder neue Feature auf Basis der Daten zu erstellen, wie beispielsweise die Hauptskomponenten bei der Hauptkomponentenanalyse oder die bestehenden Merkmale der Daten als Feature zu betrachten [25].

Der Schritt der Feature-Selektion entspricht daraufhin der Auswahl der Feature, welche eine bessere Performance auf den angestrebten Algorithmen erreichen, da nur die für den Anwendungsfall relevanten Informationen vorliegen. Zudem kann durch die Reduktion der Merkmale ein besseres Datenverständnis geschafft werden [25].

Bei der Herzschlagklassifikation auf Grundlage von EKG-Daten wird ein Feature immer dem gesamten Herzschlag zugeordnet. Es werden in vielen Studien zwei grundsätzlich zu unterscheidende Feature im Schritt der Fearure-Konstruktion verwendet [37]: Morphologische Feature und abgeleitete Feature. Die morphologischen Feature beschreiben den Herzschlag basierend auf der Beobachtung des Signals selbst, wie anhand von Tabelle 2.5 zu entnehmen ist.

Zur Konstruktion der morphologischen Feature ist keine Anwendung einer bestimmten Methode notwendig, sondern es wird lediglich das Vorhandensein der Rohdaten vorausgesetzt.

Die abgeleiteten Feature werden von dem EKG-Signal berechnet und beschreiben somit nicht die gemessenen Daten des EKGs direkt, sondern andere Kennzahlen, die sich aus den Daten ergeben oder einer anderen Repräsentationsebene entsprechen. Zur Konstruktion dieser abgeleiteten Feature haben sich diverse Methoden in der Literatur etabliert [37]. Aufgrund der sehr guten Ergebnisse in Ye et al. [74] stützen sich die Methoden, die im Rahmen dieser Arbeit verwendet werden, auf die Unabhängigkeitsanalyse und die PCA. Diese Methoden werden in weiteren Unterkapiteln genauer erläutert. Zudem wird in Ye et al. [74] auch die DWT genutzt, auf deren ausführliche Beschreibung an dieser Stelle jedoch verzichtet wird, da sie bereits in Ka-

Beispiele für morphologische Feature	
a)	Beispielpunkte des EKGs (beispielweise aus den Komplexen)
b)	Die maximale Amplitude des Herzschlags
c)	Die minimale Amplitude des Herzschlags
d)	Die Summe aus positivem und absolut negativem Bereich im QRS-Komplex
e)	Die Dauer des QRS-Komplexes
f)	Die QRS-Steigungsgeschwindigkeit, berechnet für das Zeitintervall zwischen dem Einsetzen des QRS-Komplexes und dem ersten Peak oder zwischen dem ersten und zweiten Peak
g)	Normalisiertes Signal

Tabelle 2.5: Beispiele für morphologische Feature
Quelle: In Anlehnung an Li et al., 2020 [37]

pitel 2.3.2 erfolgt ist. Ein Überblick über die verschiedenen Methoden mit bespielhaften, daraus resultierenden Feature ist durch Tabelle 2.6 gegeben.

Vektorkardiographie Vektor (VCG)	
Erfassung des räumlichen Spannungsverlaufs der elektrischen Herzaktivität in Form eines Vektorkardiogramms [10, 26]. Die EKG-Komplexe (T-/P-Welle und QRS-Komplex) werden in Form von Vektorschleifen visualisiert.	**Feature:** a) VCG-Amplitude b) VCG-Sinuswinkel c) VCG-Kosinuswinkel
Dynamische Zeitnormierung (engl. Dynamic Time Warping (DTW))	
Mithilfe von DTW kann mit bestimmten Einschränkungen eine optimale Ausrichtung zwischen zwei gegebenen (zeitabhängigen) Sequenzen gefunden werden [49]. Im Zusammenhang mit dem EKG lässt sich beispielsweise die DTW-Distanz des Herzschlags zum medianen Herzschlag der Aufzeichnung berechnen [37].	**Feature:** a) DTW-Distanz

Diskrete Wavelet Transformation (DWT)	
Die DWT beschreibt eine Zeit-Frequenz-Transformation, welche das Signal in eine endliche Menge von Untersignalen, repräsentiert als DWT-Koeffizienten, zerlegt [37]. Jedes Untersignal ist einem spezifischen Frequenzbereich zuzuordnen, was nicht nur die Rauschentfernung, sondern auch die Feature-Konstruktion ermöglicht, da in den spezifischen Koeffizientenbändern, wie D4 und D5, die Herzschlagwellen deutlicher hervorkommen. Die DWT ist in Kapitel 2.3.2 genauer erläutert.	**Feature:** a) Die positive Amplitude des R-Peaks, der T-Welle auf der vierten Dekomposition der DWT. b) Die negative Amplitude des R-Peaks auf der vierten Dekomposition der DWT. c) DWT-Koeffizienten d) Nullstellen
Unabhängigkeitsanalyse (engl. Independend Component Analysis (ICA))	
Erstellung von unabhängigen Komponenten des Signals [37]. Eine genauere Beschreibung ist im Unterkapitel *Unabhängigkeitsanalyse* 2.5 dargestellt.	**Feature:** a) ICA-Komponenten
Dual Tree Complex Wavelet Transform (DTCWT)	
Die DTCWT wird eingesetzt, um die Verschiebungsvarianz mit in der Wavelet-Zerlegung zu berücksichtigen [37]. Liegt eine Zeitverschiebung des Herzschlags vor, so ändern sich die DWT-Koeffizienten signifikant. Zur Überwindung des Problems werden statt einem, zwei Sätze von Filtern eingesetzt.	**Feature:** a) Fourier-Spektrum
Empirical Mode Decomposition (EMD)	
Zerlegung des Signals in Sub-Signale - sogenannte IMFs - [75]. Eine genauere Erklärung der Methode ist in Kapitel 2.3.3 zu finden.	**Feature:** a) IMF-Probenentropie b) IMF-Variationskoeffizient c) IMF-Singulärwerte d) IMF-Leistungsbandbreite
Hauptkomponentenanalyse (engl. Principal Component Analysis (PCA))	
Die PCA wird genutzt, um die Anzahl an Feature zu reduzieren. Sie erstellt über Linearkombinationen neue Feature, die möglichst viel Varianz in den Daten erklären können. Eine genauere Erklärung der Methode ist im Unterkapitel *Hauptkomponentenanalyse* 2.5 zu finden.	**Feature:** a) PCA-Komponenten

Eigenvektormethoden	
Eigenvektormethoden werden zum Schätzen der Frequenzen und der Leistung von rauschverfälschten Signalen verwendet [37]. Die Methoden führen eine Eigenzerlegung der Korrelationsmatrix des rauschverfälschten Signals durch.	**Feature:** a) Pisarenko PSD b) MUSCI PSD c) Minimum-Norm PSD

Tabelle 2.6: Übersicht über die Methoden zur Konstruktion von abgeleiteten Feature

In einigen Studien wird eine etwas andere Differenzierung der Feature bevorzugt. Es wird zusätzlich zu den morphologischen Feature der Begriff der dynamischen Feature eingeführt, welche die Morphologie eines Herzschlags von dem dynamischen Zusammenspiel mehrerer Herzschläge differenzieren [74]. Dabei wird als Morphologie im Gegensatz zur vorherigen Definition nicht nur die ursprüngliche Repräsentation definiert, sondern jegliche Repräsentation oder Teilrepräsentation des Herzschlags, welche die Gestalt wiedergibt. Dementsprechend würde eine DWT-Repräsentation als Morphologie gezählt werden, während RR-Intervallinformationen die Dynamik mehrerer Herzschläge beschreibt. Diese Unterscheidung basiert auf dem Nutzen der Feature zur Anomalieerkennung. Über morphologische Feature können Abweichungen der Gestalt innerhalb eines Herzschlags gefunden werden, während dynamische Feature einen Hinweis auf Unregelmäßigkeiten in Bezug auf mehrere Herzschläge, wie eine unregelmäßige Herzfrequenz, geben können.

Ein gutes Beispiel ist von Ye et al. [74] durch die Abbildung 2.11 gegeben.

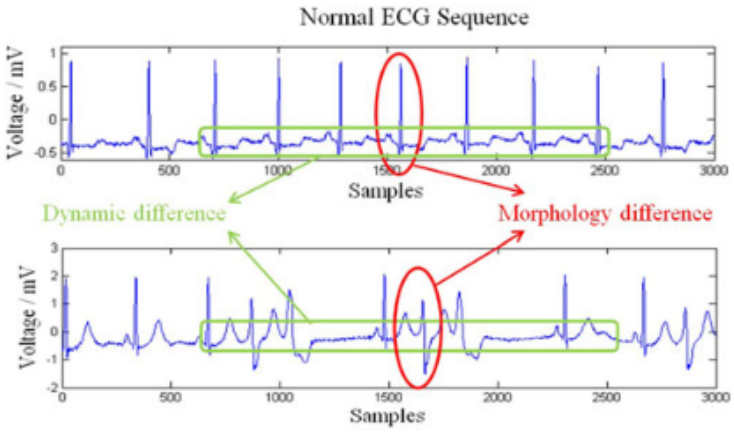

Abbildung 2.11: Differenzierung zwischen morphologischen und dynamischen Feature
Quelle: Ye et al., 2020 [74]

Unabhängigkeitsanalyse (ICA)

Das Ziel der Unabhängigkeitsanalyse ist das Finden einer linearen Abbildung von nicht-normalverteilten Daten, in der die gefundenen Komponenten so statistisch unabhängig wie möglich voneinander sind [65]. Ein oft zitiertes Beispiel zur Erläuterung der ICA ist das sogenannte *Cocktail-Party-Problem*, welches durch Abbildung 2.12 dargestellt wird.

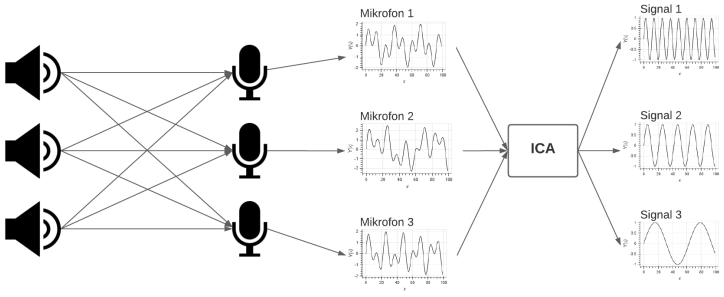

Abbildung 2.12: Cocktail-Party-Problem zur Erläuterung der ICA
Quelle: Eigene Darstellung

In der Abbildung 2.12 werden die Geräusche von drei Lautsprechern über drei Mikrofone aufgezeichnet. Das Ergebnis ist, dass jedes Mikrofon eine Mischung aus den drei Geräuschen aufzeichnet, die sich durch den unterschiedlichen Stand im Raum unterscheiden. Die ICA entmischt diese Aufzeichnungen und erhält als Komponenten jeweils die Originalsignale der Lautsprecher. Sie ist daher eine Verarbeitungsmethode zum Trennen unabhängiger Quellen, die linear in mehreren Sensoren gemischt sind. Dabei kann sowohl davon ausgegangen werden, dass Rauschen das gemischte Signal zusätzlich überlagert als auch, dass kein Rauschen vorliegt [32].

Im mathematischen Kontext bedeutet dies: Wenn $x_1(t), x_2(t), ..., x_n(t)$ beobachtete Zufallsvariablen darstellen, wird angenommen, dass es sich um eine Mischung unabhängiger Komponenten handelt, gegeben durch [36]:

$$x(t) = \mathbf{A} \cdot s(t) \tag{2.37}$$

Die Matrix $\mathbf{A}$ wird als Mischmatrix für die nichtgaußschen, unabhängigen Komponenten $s(t)$ bezeichnet. Dargestellt ist im Weiteren die Variante, bei der kein Rauschen das gemischte Signal überlagert. Ziel der ICA ist es nun $s(t)$ unter Verwendung von $x(t)$ zu schätzen, wobei zwei Bedingungen erfüllt sein müssen [36]:

1. Alle unabhängigen Komponenten $s(t)$ müssen statistisch unabhängig sein.

2. Die unabhängigen Komponenten $s(t)$ unterliegen einer nicht-gaußschen Verteilung.

Zur Durchführung der ICA wird die Mischmatrix **A** geschätzt und daraufhin die Inverse von **A** berechnet, welche als Entmischungsmatrix bezeichnet wird und definiert wird als [36]:

$$\mathbf{W} = \mathbf{A}^{-1} \tag{2.38}$$

Die unabhängigen Komponenten können daraufhin über folgende Gleichung berechnet werden [36]:

$$s(t) = \mathbf{W} \cdot x(t) \tag{2.39}$$

Hauptkomponentenanalyse (PCA)

Die Hauptkomponentenanalyse ist eine Methode, die zur Dimensionsreduktion eingesetzt wird. Dadurch wird der Variablensatz durch eine kleinere Anzahl repräsentativer Variablen dargestellt [33]. Die repräsentativen Variablen stellen Linearkombinationen der Zufallsvariablen dar, die so gewählt werden, dass sie zueinander unkorreliert sind und die Varianz maximiert wird. Ziel ist also das Finden von einem Satz repräsentativer Variablen, die möglichst viel Varianz der Daten erklären [33]. Die Herleitung der PCA basiert auf der Herleitung von Holland [31].

Abbildung 2.13 zeigt die zwei Variablen $x1$ und $x2$. Die Ausprägungen der Beobachtungen werden durch einen Scatterplot visualisiert. Beide Variablen haben ungefähr dieselbe Varianz und sind stark mit eineinander korreliert. Wird nun ein Vektor durch die Längsachse der Punktwolke und ein Vektor im rechten Winkel zum ersten Vektor geführt, wobei beide Vektoren den Schwerpunkt der Daten durchlaufen, ensteht das in Abbildung 2.13 dargestellte Bild. Die gefundenen Vektoren werden *Hauptkomponenten* oder im Englischen *Principal Components* genannt.

Zur Berechnung der Hauptkomponenten für eine Datenmatrix **X** mit p Variablen und n Beobachtungen werden die Daten zunächst auf die Mittelwerte zentriert. Dies dient der Sicherstellung, dass sich die Datenwolke auf den Ursprung der Hauptkomponenten konzentriert und zudem weder die räumlichen Beziehungen noch die Abweichungen entlang der Variablen beeinflusst werden. Die erste Hauptkomponente Y_1 ergibt sich aus der Linearkombination der Variablen $X_1, X_2, ..., X_p$, sodass gilt:

$$Y_1 = a_{11}X_1 + a_{12}X_2 + ... + a_{1p}X_p \tag{2.40}$$

In Matrixnotation entspricht dies:

$$Y_1 = \mathbf{a}_1^T \cdot \mathbf{X} \tag{2.41}$$

Die erste Hauptkomponente wird so berechnet, dass sie die größtmögliche Varianz der Daten berücksichtigt. Damit die Gewichte nicht zur Varianzvergrößerung eingesetzt werden, wird die Nebenbedingung definiert, dass die Summe der quadratischen Gewichte Eins entspricht:

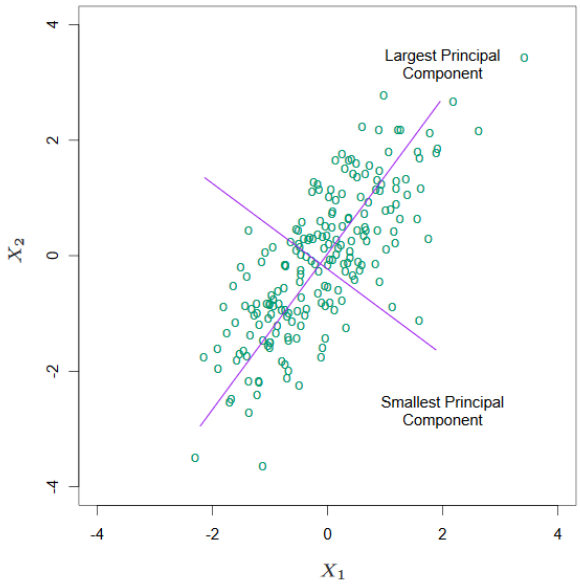

Abbildung 2.13: Beispiel von zwei Hauptkomponenten für Daten im zweidimensionalen Raum
Quelle: Hastie et al., 2009 [27]

$$a_{11}^2 + a_{12}^2 + ... + a_{1p}^2 = 1 \tag{2.42}$$

Die Berechnung der zweiten Hauptkomponente erfolgt gleichermaßen mit der zusätzlichen Nebenbedingung, dass sie nicht mit der ersten Hauptkomponente korreliert, d.h. orthogonal zueinander stehen und die nächsthöhere Varianz berücksichtigt:

$$Y_2 = a_{21}X_1 + a_{22}X_2 + ... + a_{2p}X_p \tag{2.43}$$

Insgesamt können p Hauptkomponenten auf diese Weise berechnet werden, sodass die Summe der Varianzen der p Hauptkomponenten der Summe der Varianz der p Variablen entspricht, sodass gilt:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{A} \tag{2.44}$$

Zur Dimensionsreduktion werden daraufhin die Hauptkomponenten als Feature ausgewählt, die bereits einen Großteil der Varianz erklären [74].

2.6 METHODEN ZUR HERZSCHLAGKLASSIFIKATION

Die Feature aus der Feature-Konstruktion und -Selektion bilden die Basis zum Trainieren eines Modells. Dies soll in der Lage sein zwischen abnormalen und normalen Herzschlägen zu differenzieren. Innerhalb dieses Kapitels wird auf die Algorithmen, die sich bei der EKG-Anomalieerkennung bewährt haben, im Speziellen auf die SVM, welche bei Ye et al. [74] besonders gute Ergebnisse erreicht hat, eingegangen.

Zur EKG-Anomalieerkennung werden insbesondere Clustering-basierte Methoden, traditionelle Machine-Learning-Algorithmen und Deep-Learning-Methoden eingesetzt [37].

Aus den Clustering-basierten Verfahren ist das K-mean Clustering eine oft gewählte Methode, welche auf Basis der Euklidischen Distanz die einzelnen Herzschläge klassifiziert. Veeravalli et al. [70] erreichten über diesen Ansatz auf der MIT-BIH Datenbank [40] und der European ST-T Datenbank eine Sensitivität von 97,1% und eine Spezifität von 99,5%.

Aus dem Bereich der traditionellen Machine-Learning-Verfahren finden die Methoden der K-Nächste-Nachbarn (KNN), Lineare Diskriminanzanalyse (LDA), Quadratische Diskriminanzfunktion (QDF) und der SVM Anwendung. Eine Übersicht über einige bisher publizierte Ergebnisse ist durch Tabelle 2.7 gegeben.

Methode	**Anzahl Typen**	**Accuracy**	**Referenz**
KNN	5	96%	Christov et al., 2006 [15]
LDA	5	97%	De Chazal et al., 2006 [57]
QDF	5	94%	Llamedo et al., 2011 [39]
One-Class SVM	2	87,89%	Li et al., 2012 [38]
Multi-Class SVM	16	99,77%	Ye et al., 2012 [74]

Tabelle 2.7: Übersicht über einige Ergebnisse von Publikationen zur Klassifikation von Herzschlägen auf dem MIT-BIH-Datensatz mithilfe traditioneller Machine-Learning-Verfahren

Viele Publikationen basieren auf der Anwendung der traditionellen Methoden, da sie keine beträchtliche Menge an Trainingsdaten und viel Rechenkapazität voraussetzen [37]. Durch die Entwicklung der GPUs haben sich jedoch auch die Deep-Learning-Methoden bewährt, weshalb diese vermehrt Anwendung finden und sich ebenfalls für die Anomalieerkennung auf EKG-Daten eignen [37].

Innerhalb dieser Arbeit stehen keine GPUs zur Verfügung, weshalb auf den Ansatz der traditionellen Machine-Learning-Verfahren zurückgegriffen wird. Da die SVM in diesem Bereich die besten Forschungsergebnisse aufzeigte, wird im Folgenden näher auf diees Verfahren eingegangen.

2.6.1 *Support Vector Machine (SVM)*

Die Support Vector Machine (SVM) stellt eine der bisher erfolgreichsten entwickelten Data-Mining-Methoden dar [8, 33]. Sie wird regelmäßig in der Industrie und in der Forschung eingesetzt. Liegen jedoch genug Trainingsdaten und eine entsprechende Hardware vor, so wird sie mittlerweile oft durch Deep Neural Networks ersetzt. Dieses Kapitel basiert auf der Herleitung von Ben-Hur et al. [6].

Lineare SVM

Die Grundidee der SVM ist die Konstruktion einer Hyperebene. Zur Darstellung dieser Hyperebene sind einige Definitionen notwendig. Die Daten für ein Zwei-Klassen-Problem bestehen aus Objekten, die mit einem von zwei Labels gekennzeichnet sind. Zur Vereinfachung wird angenommen, dass die Labels $+1$ (positive Beispiele) und -1 (negative Beispiele) entsprechen. Zusätzlich sei $\mathbf{x}$ ein Vektor mit M Komponenten x_j, mit $j = 1, ..., M$ (ein Punkt im M-dimensionalen Vektor-Raum). Die Notation $\mathbf{x}_i$ beschreibt den i-ten Vektor in einem Datensatz $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, wobei y_i dem Label von $\mathbf{x}_i$ entspricht und n die Anzahl an Input-Daten beschreibt.

Ein Schlüsselkonzept, das für die Definition eines linearen Klassifikators erforderlich ist, ist das Skalarprodukt zwischen zwei Vektoren $\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{j=1}^{M} w_j x_j$. Die Hyperebenenfunktion wird mithilfe des Skalarprodukts definiert:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{2.45}$$

$f(\mathbf{x})$ gibt den *Score* für die Input-Daten $\mathbf{x}$ an und entscheidet die Klassifikation. $\mathbf{w}$ wird als *Gewichtsvektor* und b als *Bias* bezeichnet. Die Punkte, die die Gleichung

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0, \tag{2.46}$$

erfüllen, entsprechen der Hyperebene. Abbildung 2.14 zeigt zwei Beispiele zur Wahl einer solchen Hyperebene. Beide Beispiele separieren die gelabelten Daten in Bezug zur Klassenzuordnung, welche über die Farbgebung symbolisiert wird. a) und b) zeigen beide eine Ausführung mit unterschiedlichem *Gewichtsvektor* $\mathbf{w}$ und *Bias* b.

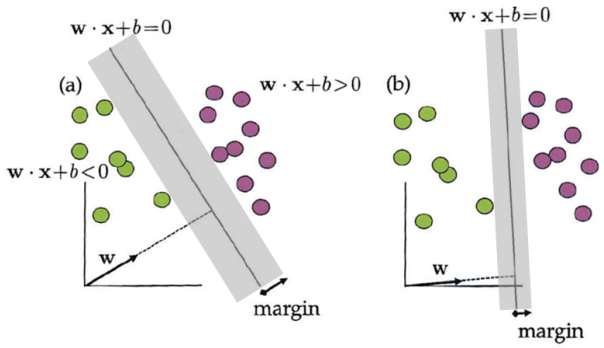

Abbildung 2.14: Der SVM-Klassifizierer konstruiert eine Hyperebene, die optimal die gelabelten Daten separiert. Die Farben entsprechen einer Klassenzuordnung. a) Zeigt eine Hyperebene mit maximaler Margin. b) Zeigt eine Hyperebene, die dieselbe Klassienseperation darstellt, aber keine maximale Margin aufweist.
Quelle: Brunton und Kutz, 2019 [8] (Die Abbildung wurde angepasst.)

Das Optimierungsproblem der SVM bezieht sich jedoch nicht nur darauf, die Daten so zu separieren, dass die geringsten Fehler bei der Klassenzuord-

nung entstehen, sondern auch die sogenannte *Margin* zu maximieren. Die Margin bezeichnet den minimalen Abstand zwischen der Beobachtung, die auf jeder Seite am nächsten an der Hyperebene liegt, und der Hyperebene. In Abbildung 2.14 ist die Margin auf jeder Seite der Hyperebene in Grau hinterlegt. Die Vektoren, die die Grenzen der Margin bestimmen, werden *Support-Vektoren* genannt.

Da sich eine Skalierung von $\mathbf{w}$ und b nicht auf die Distanz $\mathbf{x}$ zur Hyperebene auswirkt, wird $f(\mathbf{x})$ so skaliert, dass die Margin $\frac{1}{\|\mathbf{w}\|}$ entspricht.

Die sogenannte *Hard Margin* SVM entspricht einem Klassifikator für Daten die linear separierbar sind. Zur Berechnung von $\mathbf{w}$ und b ist die Lösung des nachfolgenden Optimierungsproblems notwendig:

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}\rangle + b) \geq 1, \text{für } i = 1, ..., n \end{aligned} \tag{2.47}$$

Die Minimierung von $\|\mathbf{w}\|^2$ ist äquivalent zur Maximierung der Margin. In der Realität sind jedoch oft Daten vorzufinden, welche nicht linear separierbar sind, oder durch die Missklassifikation von einzelnen Punkten die Margin deutlich vergrößert werden kann. Studien zeigen, dass eine größere Margin in den meisten Fällen die Modellperformance steigert, im Vergleich zur Hard Margin SVM, weshalb sich die sogenannte *Soft Margin* SVM etabliert hat. Sie nutzt die *Schlupfvariablen* ξ, um Fehler zu erlauben. Es ergibt sich ein neues Optimierungsproblem:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}\rangle + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \text{für } i = 1, ..., n \end{aligned} \tag{2.48}$$

Die Konstante $C > 0$ legt als Kostenfaktor die Wichtigkeit zwischen der Maximierung der Margin und der Minimierung der Falschklassifikationen fest.

Nicht-lineare SVM

Lineare Klassifikatoren haben zwar den Vorteil der leichten Interpretation, sind jedoch zu restriktiv für Daten, die in einem hochdimensionalen Raum eingebettet sind. Zur Erstellung komplexerer Klassifikationskurven wird der Funktionsbereich der SVM erweitert. Es werden nicht-lineare Feature eingeführt und mithilfe dieser Feature Hyperebenen in dem neuen Raum definiert. Das heißt, die Daten werden einem nicht-linearen, höherdimensionalen Raum zugeordnet

$$\mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x}). \tag{2.49}$$

$\mathbf{\Phi}(\mathbf{x})$ werden als neue Beobachtungen der Daten bezeichnet. Mithilfe von $\mathbf{\Phi}(\mathbf{x})$ kann die SVM nun die Hyperebenen lernen, aus denen eine optima-

le Trennung der Daten in einem neuen Raum hervorgeht, sodass die neue Hyperebenenfunktion gegeben ist durch:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}) \rangle + b \tag{2.50}$$

Die Funktion definiert die Labels $\mathbf{y}_i \in \{\pm 1\}$ für jeden Datenpunkt $f(\mathbf{x}_i)$.

Die Idee den Funktionsraum zur Definition der Daten $\mathbf{x}$ zu vergrößern, hat sich in der Klassifikation der Daten als ein außergewöhnlich leistungsstarkes Mittel herausgestellt [8]. Als Beispiel können zweidimensionale Daten $\mathbf{x} = (x_1, x_2)$ als Polynome definiert werden, um den Funktionsraum zu erweitern.

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1, x_2, x_1^2 + x_2^2). \tag{2.51}$$

Mithilfe des Polynoms zur Definition von z_3 können die Daten $\mathbf{x}$ nun in einem dreidimensionalen Raum dargestellt werden und einfacher Hyperebenen zur Trennung der Klassen bestimmt werden. Abbildung 2.15 zeigt eine Visualisierung der Variablen x_1, x_2 sowie z_1, z_2 und z_3.

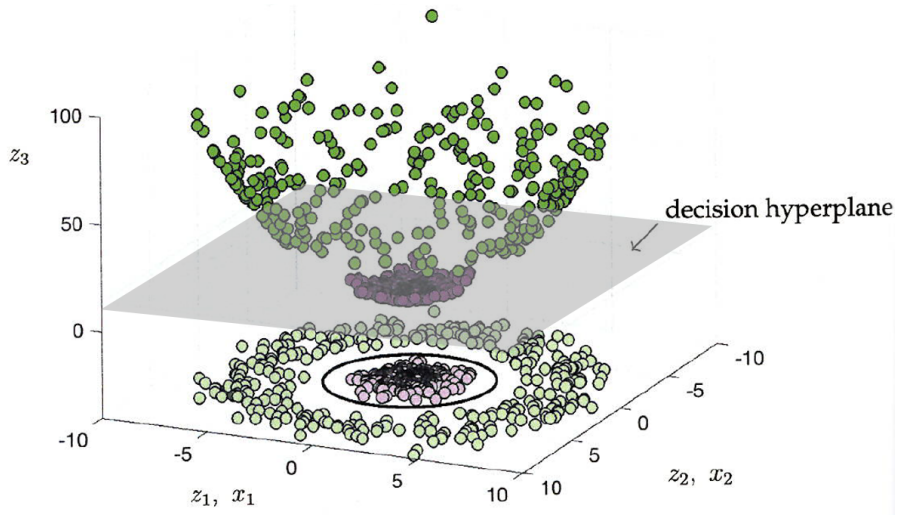

Abbildung 2.15: Beispielhafte Darstellung der Variablen $(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1, x_2, x_1^2 + x_2^2)$ zur Visualisierung des Nutzens einer nicht-linearen SVM. Unten sind die Daten im zweidimensionalen Raum angegeben. Eine lineare Trennung ist nicht möglich. Nach der Erhöhung des Funktionsraums durch die Definition eines polynomialen Feature kann eine Hyperebene definiert werden, welche die beiden Klassen eindeutig voneinander trennen kann.
Quelle: Brunton und Kutz, 2019 [8] (Die Abbildung wurde angepasst.)

Durch Einführung des höherdimensionalen Funktionsraums ist nun eine lineare Trennung über eine Hyperebene möglich, während die zuvorige Trennung lediglich über einen Kreis erfolgen konnte. Dieser ist jedoch nicht mit einer Hyperebene darstellbar. Die zugrunde liegende Optimierungsfunktion bleibt dieselbe. Mithilfe der Funktion $\mathbf{\Phi}(\mathbf{x}_i)$ können nun die Daten in einem höheren Funktionsraum definiert werden, um so eine bessere Klassifikation zu erreichen.

Kernel-Methoden der SVM

Trotz der Erhöhung des Funktionsraums führt der nicht-lineare Klassifikator zu einer rechnerisch unlösbaren Optimierung. Das Berechnen der Vektoren $\mathbf{w}$ ist sehr teuer und es ist zudem nicht sichergestellt, dass diese überhaupt im Hauptspeicher dargestellt werden kann. Mithilfe des sogenannten *Kernel-Tricks* ist es möglich die Rechenressourcen deutlich zu verringern, indem der Vektor $\mathbf{w}$ dargestellt wird durch:

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i \tag{2.52}$$

α_i sind Gewichtungsparameter. Die Punkte $\mathbf{x}_i$, für die $\alpha < 0$ ist, entsprechen den *Support Vekoren*. Der Vektor $\mathbf{w}$ wird dadurch in einen beobachteten Satz von Funktionen erweitert, wodurch die Hyperebenenfunktion verallgemeinert folgendermaßen dargestellt werden kann:

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}) \rangle + b \tag{2.53}$$

Die *Kernel-Funktion* kann daraufhin definiert werden als:

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{x}') \rangle \tag{2.54}$$

Durch die Kernel-Funktion ist es möglich, in einem hochdimensionalen, impliziten Merkmalsraum zu arbeiten, ohne die Koordinaten der Daten in dem Raum selbst berechnen zu müssen. Stattdessen wird lediglich das innere Produkt aller Datenpaare im Merkmalsraum berechnet. Die zwei am häufigsten genutzen Kernel-Funktionen sind:

$$\text{Radial basis functions (RBF): } K(\mathbf{x}, \mathbf{x}') = exp\left(-y\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{2.55}$$

$$\text{Polynomial kernel: } K(\mathbf{x}, \mathbf{x}') = \left(\langle \mathbf{x}, \mathbf{x}' \rangle + k\right)^d \tag{2.56}$$

d entspricht dem Grad des zu berücksichtigenden Polynoms und y der Breite des Gaußschen Kernels, der den Abstand zwischen einzelnen Datenpunkten zur Klassifikationslinie misst. k beschreibt eine Konstante, welche oft als 0 oder 1 gewählt wird. Die Konstruktion höherdimensionaler Räume durch Beobachtungen, die mit Kernel-Funktionen generiert werden, entspricht einer der wichtigsten Eigenschaften der SVMs. Es ermöglicht eine nicht-lineare Klassifikation mit geringem Rechenaufwand.

3

DATENGRUNDLAGE

Bei den vorliegenden Daten handelt es sich um die Monitoring-Daten der Intensivstation und der Schlaganfallstationen. Der Regelbetrieb der Datenaufzeichnung wurde erst zum Ende der Bearbeitungszeit gestartet, jedoch gab es auf beiden Stationsgruppen bereits einen Testdurchlauf, auf welchem die weitere Arbeit basiert. Auf der Intensivstation wurde insgesamt ein Zeitraum von vier Wochen aufgezeichnet. Für die Schlaganfallstationen liegen Daten von zehn Tagen vor.

Insgesamt wurden über 100 mögliche Parameter aufgezeichnet, welche je nach Patient variieren. Einige der Parameter werden bei fast jedem Patienten überwacht, weshalb sie von stärkerem Interesse sind. Aus dieser Teilmenge wurden die Parameter seitens der Charité priorisiert. Auf den daraus hervorgegangenen wichtigsten Parameter - das Elektrokardiogramm - konzentrieren sich die Untersuchungen dieser Arbeit. Insgesamt liegen Zeitreihen der verschiedenen Parameter von 133 Patienten vor, dabei handelt es sich bei 61 Patienten um Intensivstationspatienten und bei 72 um Patienten der Schlaganfallstationen. Die durchschnittliche Länge der Aufenthaltsdauer eines Patienten entspricht 132 Stunden (5,5 Tage). Das Minimum liegt bei einer Stunde, das Maximum bei 749 Stunden (31 Tage). Insgesamt liegt eine Datenmenge von 15.077 Stunden, was ungefähr 628 Tagen entspricht, des Parameters EKG vor.

An die EKG-Parameter sind einige technische Gegebenheiten geknüpft, welche durch die Tabelle 3.1 dargelegt werden.

Technische Gegebenheiten
• 512 Werte pro Sekunde (512 Hz-Aufnahme)
• Aufzeichnung von mindestens zwei EKG-Ableitungen
• Die aufgezeichneten Ableitungen variieren nach Vorliebe des behandelnden Arztes
• Die zeitliche Zuordnung der aufeinander folgenden Werte wird alle fünf Sekunden aktualisiert

Tabelle 3.1: Technische Gegebenheiten der der EKG-Aufzeichnung

Mit jedem zu einem Zeitstempel übermittelten Wert werden noch 30 weitere Informationen mitgegeben. Wichtige Zusatzinformationen sind zum Beispiel die pseudonymisierte Patienten-ID, die eingestellten Alarmgrenzen, die Einheit des Wertes, das Label in Bezug auf den Parameter und das Sub-Label, falls eine weitere Unterteilung des Labels notwendig ist. Eine weitere Datenquelle ist das Alarmsystem, welches noch weitere Informationen

bereitstellt. Die Alarmdaten-Tabelle ordnet jedem ausgelösten Alarm eine Patienten-ID, einen Zeitstempel, eine Beschreibung des Alarms in Form eines Labels, einer Alarmkategorie und eine Alarmschwere zu. Die Kategorie entspricht einer Einordnung, ob es sich um eine Information bezüglich des Patienten, etwas Technisches, eine Benachrichtigung, einen Status oder eine nicht zu bestimmende Kategorie handelt. Die Alarmschwere bezieht sich auf den Schweregrad des vorliegenden Alarms. Einer Asystolie ist beispielsweise ein höherer Schweregrad zugeteilt als einer einfachen technischen Störung.

Visualisierung des EKGs

Da es sich bei den hochaufgelösten EKG-Parametern um sehr große Datenmengen handelt, kann mittels Spark in Zeppelin nur ein begrenztes Zeitfenster visualisiert werden. Dies liegt daran, dass die Zeitreihen über die interaktive Python-Visualisierungsbibliothek Bokeh[1] dargestellt werden. Der Einsatz von Python-Bibliotheken bedingt eine Umwandlung von Pyspark Dataframes in Pandas Dataframes, was zur Folge hat, dass alle Daten in den RAM geladen werden [19]. Dies ist jedoch nur bis zu einer bestimmten Datenmenge möglich und auch performant umzusetzen. Insbesondere bei den hochaufgelösten Zeitreihen führt dies aufgrund der Datenmenge zu Problemen, sodass keine Zeitreihen über den gesamten Aufenthaltszeitraum dargestellt werden können. Ausschnitte lassen sich jedoch ohne Probleme visualisieren.

Der Parameter *EKG* entspricht im vorliegenden System den drei Ableitungen nach Einthoven (I, II, III) und den drei Ableitungen nach Goldberger (aVR, aVL, aVF). Zwei dieser sechs Ableitungen werden durch das System aufgezeichnet. Die restlichen vier werden mittels folgender Gleichungen berechnet.

$$\begin{aligned} -aVR &= \tfrac{I+II}{2} \\ aVL &= \tfrac{I-III}{2} \\ avF &= \tfrac{II+III}{2} \\ I+II &= III \end{aligned}$$

Tabelle 3.2: Gleichungssystem der EKG-Ableitungen

1 https://docs.bokeh.org/

Die nachfolgende Abbildung 3.1 zeigt einen Ausschnitt über 10 Sekunden der EKG-Ableitungen.

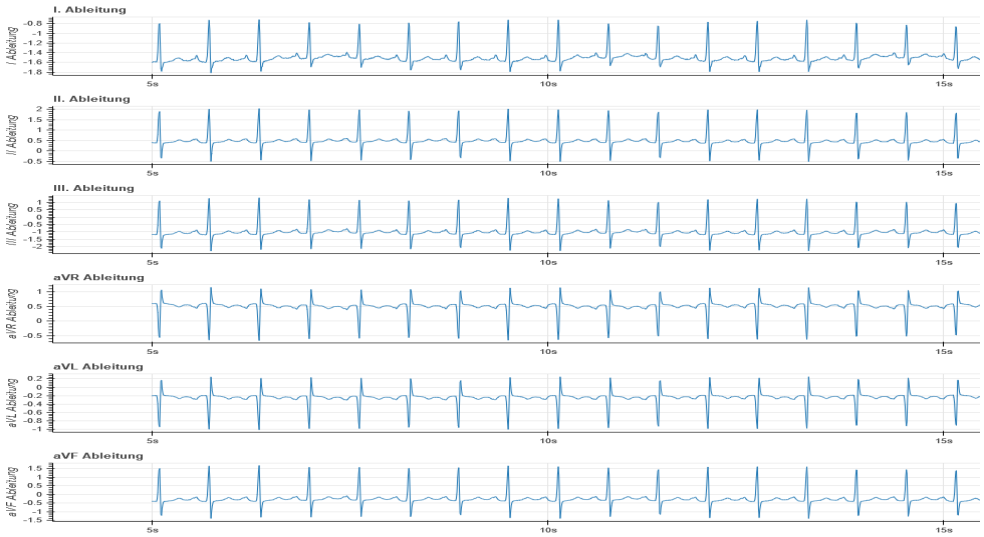

Abbildung 3.1: Visualisierung der EKG-Ableitungen nach Einthoven und nach Goldberger über 10 Sekunden
Quelle: Eigene Darstellung

Datenvorbereitung

Es werden Daten aus drei verschiedenen Systemen auf die Health Data Platform (HDP) übertragen. Die *Numerics*, welche den Daten der meisten Vitalparametern entsprechen, beinhalten alle Daten, die eine geringe Aufzeichnungsfrequenz benötigen. Die Aufzeichnungsfrequenz liegt bei diesen Parametern bei weniger als einem bis zu drei Werten pro Sekunde, wie es bei der Sauerstoffsättigung oder der Atemfrequenz der Fall ist. Die *Waves* entsprechen den Daten, die eine sehr hohe Aufzeichnungsfrequenz benötigen, wie beispielsweise das EKG oder der arterielle Bludruck. Die dritte Datenquelle ist das Alarmsystem, welche die Philips-Monitoring-Annotationen (*Alerts*) als Daten liefert. Diese beinhalten alle aufgezeichneten Alarme und die zugehörigen Systemeinstellungen.

Damit die Daten visualisiert werden können, sind einige Pre-processing-Schritte notwendig. Die *Numerics* und die *Waves* bestehen jeweils, wenn sie aus den Systemen geliefert werden, aus zwei Tabellen. Zur Nutzung der Daten müssen die beiden Tabellen miteinander über ein Schlüsselelement verbunden werden. Die Waves liegen zudem aufgrund des hohen Datenaufkommens komprimierter vor, indem die Daten immer in Fünf-Sekunden-Abschnitte in Binär-Listen gespeichert werden, sodass diese alle denselben Zeitstempeln und denselben Eigenschaften zugeordnet werden. Zur Analyse und Visualisierung ist die Umwandlung der Daten in das Dezimalsystem und eine Zuordnung der genauen Zeitstempel notwendig.

4

METHODENAUSWAHL ZUR ELIMINATION TECHNISCHER ARTEFAKTE AUF BASIS DES CHARITÉ-DATENSATZES

In diesem Schritt geht es darum, Methoden zur Elimination technischer Artefakte in den EKG-Zeitreihen zu vergleichen. Es ist festzustellen, dass je höher die Qualität der Daten ist, desto höhere Erfolge sind im Zuge der Anomalieerkennung zu verzeichnen. Die hohe Qualität der Daten geht mit möglichst wenig verrauschten Daten einher. Da sich noch kein einheitliches Vorgehen zur Entrauschung von EKG-Daten etabliert hat, werden drei Verfahren miteinander verglichen: Sparse Signal Decomposition, DWT und CEEMDAN. Alle drei Methoden zeigten in Studien bereits hohes Potential die Daten erfolgreich zu bereinigen [50, 61, 73]. Die ausgewählten drei Ansätze werden implementiert, auf einem Testdatensatz evaluiert und das beste Ergebnis als Entrauschungsprozedur für die Daten vor der anschließenden Anomalieerkennung ausgewählt. Da sich bei der Implementierung gezeigt hat, dass die Bereinigung von Muscle Artefacts oft zu schlechten entrauschten Zeitreihen führte, wird im Folgenden lediglich auf die Elimination von Baseline Wander und Power-Line-Interferenz eingegangen. Dies liegt daran, dass ein Muscle Artefact keiner bestimmten Frequenzspanne zugeordnet werden kann, die sich von dem EKG-Signal abhebt, sodass eine automatische Bereinigung zu schnell Verzerrungen des eigentlichen Signals hervorruft.

4.1 ERSTELLUNG EINES TESTDATENSATZES

Die Erstellung eines Testdatensatzes dient der Evaluation der drei Methoden, damit eine Entrauschungsprozedur für die Daten ausgewählt werden kann. Da für EKG-Signale mit einem beobachteten Artefakt keine Version ohne Artefakt vorliegt, werden Artefakte simuliert und auf unverzerrte EKG-Signale addiert. Zur Definition des Testdatensatzes werden daher 30 verschiedene 10-Sekunden-Abschnitte in den Daten gesucht, die möglichst keine technischen Artefakte aufweisen, sodass ein Vergleich zwischen einer Zeitreihe, bei dem Artefakte eliminiert wurden und derselben Zeitreihe ohne Artefakt ermöglicht wird. Die Auswahl der Zeitreihen basiert auf verschiedenen Eigenschaften, die den Testdatesatz repräsentieren soll. Zunächst wird darauf geachtet, dass jede der sechs Ableitungen zu gleichen Teilen in dem Testdatensatz vertreten ist. Zudem werden Zeitreihen sowohl mit als auch ohne vorliegende medizinische Events ausgesucht, um zu garantieren, dass die Entrauschungsprozeduren die medizininschen Events nicht verzerren. Eine weitere Eigenschaft der extrahierten Zeitreihen ist, dass sie zur Hälfte aus Aufnahmen zur Tageszeit und zur anderen Häfte aus Aufnahmen zur Nachtzeit bestehen.

Damit die einzelnen Methoden verglichen werden können, werden drei Arten eines Baseline Wanders (Abbildung A.1), drei Arten von Power-Line-Interferenz (Abbildung A.2) und drei verschiedene Kombinationen aus den beiden Artefakten (Abbildung A.3) simuliert und auf jeweils 10 der Originalzeitreihen addiert, sodass $(3 + 3 + 3) \cdot 10 = 90$ Testzeitreihen entstehen. Ein Beispiel eines simulierten Artefakts ist durch Abbildung 4.1 gegeben. Abbildungen von allen simulierten Artefakten ist dem Appendix Kapitel A.1 zu entnehmen.

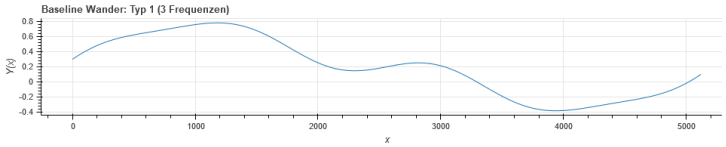

Abbildung 4.1: Simuliertes Baseline Wander zur Evaluation der Entrauschungsmethoden
Quelle: Eigene Darstellung

Durch den simulierte Testdatensatz wird ein Vergleich der Methoden ermöglicht, welcher auf Basis beobachteter verunreinigter Zeitreihen nicht möglich gewesen wäre, da dann keine Vergleichszeitreihen vorliegen.

4.2 IMPLEMENTIERUNG DER METHODEN

Insgesamt werden drei Methoden implementiert. Dazu zählen die DWT, die CEEMDAN und die Sparse Signal Decomposition. Zusätzlich wird die DWT mit zwei unterschiedlichen Basisfunktionen als Mother-Wavelet umgesetzt, sodass insgesamt vier Varianten verglichen werden. Wie durch Arumugam et al. [5], Li et al. [37] und Saxena et al. [63] zu entnehmen ist, stellen die Daubechies Wavelets der Ordnung 4 und 8 oft genutzte Varianten dar, die in Bezug auf verschiedene eingesetzte Metriken gute Ergebnisse liefern. In dieser Arbeit wird für die Implementierung das Daubechies Wavelet der Ordnung 4 herangezogen. Ein weiteres Paper von Nguyen et al. [50] vergleicht die üblich genutzten Wavelets zur Entrauschung von EKG-Daten, wobei das Discrete Meyer Wavelet (dmey) die besten Ergebnisse in Bezug auf den Signal-to-Noise Ratio (SNR) und den Mean Squared Error (MSE) erzielte, wodurch dieses ebenfalls implementiert wird. Im Weiteren Verlauf werden die Implementationen der einzelnen Methoden genauer erläutert.

Diskrete Wavelet-Transformation

Bei der Implementierung der DWT wird auf die Python-Bibliothek PyWavelets[1] zurückgegriffen und die DWT als Filterbank implementiert. Zur Implementierung der Filterbank werden acht Dekompositionslevel genutzt, sodass daraus acht Detailkoeffizienten und ein Approximationskoeffizient resultieren. Als Mother-Wavelet werden, wie bereits erläutert, die zwei Basisfunktionen db4 und dmey herangezogen. Nach der Zerlegung in die einzelnen

1 https://pywavelets.readthedocs.io/en/latest/

Koeffizienten, können diese einzelnen Frequenzspannen zugeordnet werden. Die Tabelle 4.1 zeigt diese Zuordnungen für das vorliegende 512 Hz Signal. In Kapitel 2.3.2 wurde diese Tabelle bereits vorgestellt.

Level	DWT-Koeffizienten	Frequenzbereich (in Hz)
1	$\mathbf{d_1}$	128-256
2	$\mathbf{d_2}$	65-128
3	$\mathbf{d_3}$	32.5-65
4	$\mathbf{d_4}$	16.25-32.5
5	$\mathbf{d_5}$	8.125-16.25
6	$\mathbf{d_6}$	4.063-8.125
7	$\mathbf{d_7}$	2.031-4.063
8	$\mathbf{d_8}$	1.016-2.031
8	$\mathbf{a_8}$	0-1.016

Tabelle 4.1: Frequenzbereiche der DWT-Koeffizienten der 8 Dekompositionslevel bei einem 512 Hz Signal, Quelle: In Anlehnung an Nguyen, 2020 [50]

Die Visualisierung der einzelnen rekonstruierten Koeffizienten ist in Abbildung 4.2 zu sehen. Hier ist zu erkennen, dass die ersten drei Koeffizienten feines Rauschen repräsentieren, wie es bei dem Power-Line-Interferenz-Artefakt der Fall ist. Dieses bewegt sich im Normalfall zwischen 47 und 53 Hz [34], was der Frequenzspanne des dritten Detailkoeffizienten entspricht. Da es sich bei der Visualisierung um eine beobachtete verunreinigte Zeitreihe handelt, ist auch weiteres Rauschen in den Detailkoeffizienten der höheren Frequenzen zu finden. Die Frequenzen des EKG-Signals selbst bewegen sich in niedrigeren Bereichen, wodurch die in Abbildung 4.2 zu erkennenden Ausschläge in d1 und d2 eindeutig Rauschem zuzuordnen sind. Das Baseline-Wander-Artefakt bewegt sich in einer Frequenz von Null bis Einem Hz [34], sodass der Approximationskoeffizient a8 diesem zuzuordnen ist. Zur Entrauschung werden die Koeffizienten außerhalb der erwarteten EKG-Frequenz (d1, d2, d3 und a8) auf Null gesetzt und daraufhin das Signal rekonstruiert.

Das Ergebnis nach der Entrauschung mittels der DWT ist Abbildung 4.3 zu entnehmen.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

Bei der Implementierung der CEEMDAN kann auf ein Python-Paket (PyEMD[2]) zurückgegriffen werden, sodass die Methode nicht von Grund auf implementiert werden muss. Als Denoising-Mechanismus wird ebenfalls die DWT eingesetzt, um die einzelnen IMFs zu entrauschen. Zusätzlich werden nur die IMFs in die Rekonstruktion mit einbezogen, die weniger als 1,5 Nullstellen pro Sekunde aufweisen. Der CEEMDAN kann mit verschiedenen Algorithmen

2 https://pyemd.readthedocs.io/en/latest/

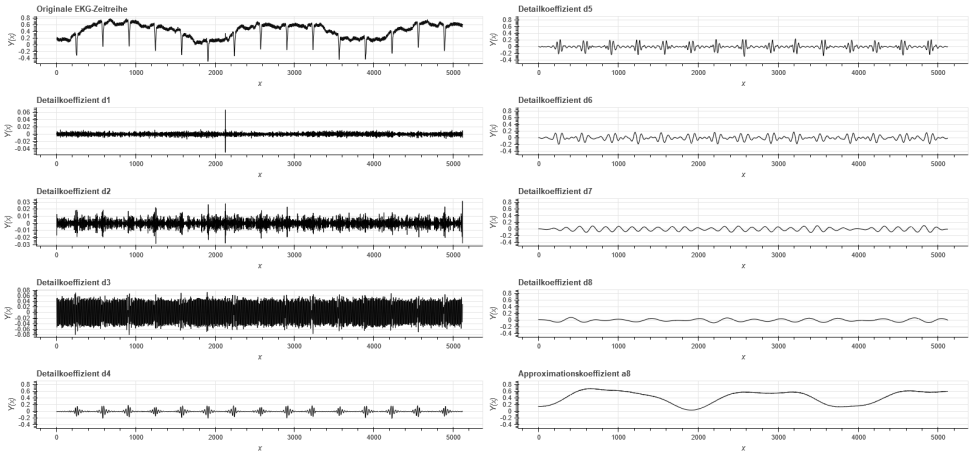

Abbildung 4.2: Rekonstruierte Zeitreihen für jedes Dekompositionslevel nach Anwendung der DWT mit dem Discrete Meyer Wavelet, Quelle: Eigene Darstellung mithilfe des Python-Pakets PyWavelets [54]

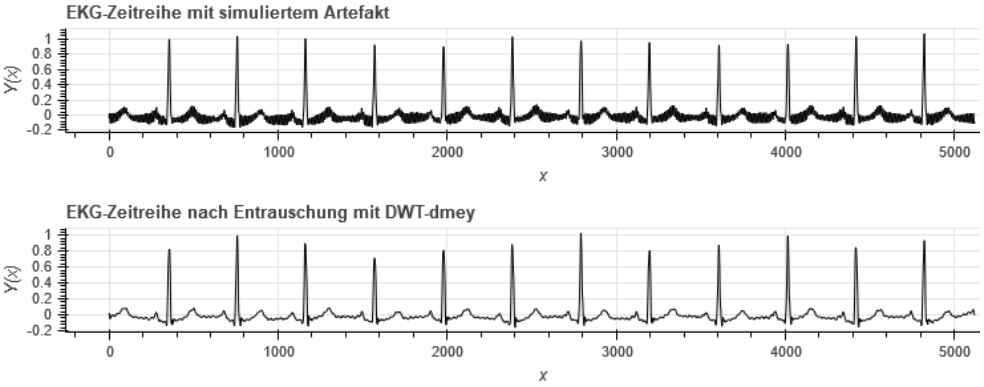

Abbildung 4.3: Oben: Zeitreihe mit simuliertem, hinzugefügtem Artefakt. Unten: Entrauschte Zeitreihe nach Anwendung der DWT mit dem Discrete Meyer Wavelet als Basisfunktion, Quelle: Eigene Darstellung

genutzt werden, wie beispielsweise Threshold-Mechanismen [21, 73]. Außerdem besteht die Möglichkeit einzelne IMFs direkt als Rauschen zu klassifizieren und von der Rekonstruktion auszuschließen, wie es oben beschrieben wird [18]. Ein in die IMFs zerlegtes Signal ist in Abbildung 4.4 zu sehen.

Ein Besipiel für eine entrauschte Zeitreihe über den CEEMDAN zeigt Abbildung 4.5.

Sparse Signal Decomposition

Für die Sparse Signal Decomposition liegt kein Python-Paket vor, sodass diese Methode vollständig implementiert werden muss. Dafür muss zunächst das Overcomplete Dictionary erstellt werden, welches sowohl Sinus- als auch Kosinuswellen als Basisfunktionen nutzt. Die Koeffizienten der Basismatrix werden dabei entsprechend der im Kapitel 2.3 beschriebenen Formeln 2.5 und 2.6 berechnet. Der Code-Auszug 4.1 zeigt diese Berechnung. Aus der Matrix werden daraufhin die zum Baseline Wander und Power-Line-Interfe-

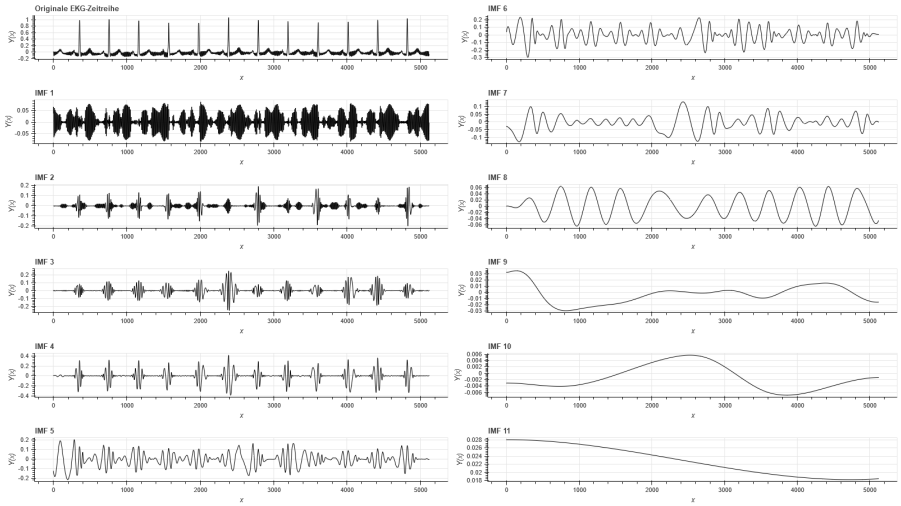

Abbildung 4.4: Ein EKG-Signal mit Power-Line-Interferenz, was in seine einzelnen IMFs zerlegt wurde.
Quelle: Eigene Darstellung mithilfe des Python-Pakets PyEMD [53]

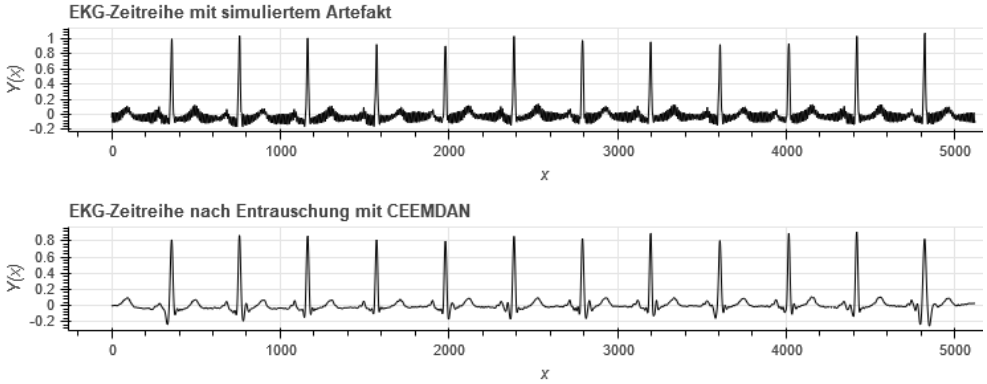

Abbildung 4.5: Oben: Zeitreihe mit simuliertem, hinzugefügtem Artefakt. Unten: Entrauschte Zeitreihe nach Anwendung des CEEMDAN-Algorithmus mit DWT, Quelle: Eigene Darstellung

renz gehörigen Frequenzen extrahiert und damit das Dictionary zur Bereiningung der beiden Artefakte erstellt.

```
#Create Sinus- & Cosinus-Transformation Matrix
s_ij = np.fromfunction(lambda i, j: np.sqrt(2/P)*(1* np.sin((np.pi*(2*j + 1)*(i
    +1))/(2*P))), (P-1,P))
last_row_s = np.array([np.sqrt(2/P)*(np.sqrt(1/2)* np.sin((np.pi*(2*j + 1)*(P+1)
    )/(2*P))) for j in range(P)])
s_ij = np.vstack([s_ij, last_row_s])
c_ij = np.fromfunction(lambda i, j: np.sqrt(2/P)*(1* np.cos((np.pi*(2*j + 1)*(i)
    )/(2*P))), (P,P))
first_row_c = np.array([np.sqrt(2/P)*(np.sqrt(1/2)* np.cos((np.pi*(2*j + 1)*(0))
    /(2*P))) for j in range(P)])
c_ij = np.vstack([first_row_c, c_ij[1:]])
#Extraction of Baseline Wander & PLI frequencies
bw_s = s_ij[:,0:20]
bw_c = c_ij[:,0:20]
pli_s = s_ij[:,900:1060]
```

```
pli_c = c_ij[:,900:1060]
#Create Phi Matrix (Combining of sine- and cosine transformation vectors)
phi_bw = np.append(bw_c, bw_s, axis=1)
phi_pli = np.append(pli_c, pli_s, axis=1)
phi = np.append(phi_bw, phi_pli, axis=1)
```

Listing 4.1: Berechnung der eindimensionalen diskreten Sinus- und Kosinustransformation und Extraktion der zu den Artefakten gehörigen Elementarwellen.

Im nächsten Schritt wird der Sparse-Vektor auf Basis der konvexen l_1-Norm-Optimierung geschätzt. Dafür wird das Python-Paket cvxpy[3] genutzt. Der Sparse-Vektor repräsentiert die Koeffizienten zur Schätzung des Baseline Wanders und der Power-Line-Interferenz. Der Code-Ausschnitt 4.2 zeigt dieses Vorgehen. Insgesamt werden 360 Koeffizienten geschätzt. 40 Koeffizienten repräsentieren mit den zugehörigen Elementarwellen das Baseline Wander. 320 Koeffizienten dienen der Schäzung der Power-Line-Interferenz. Die unterschiedliche Anzahl resultiert aus den umfänglicheren Frequenzen, die bei der Power-Line-Interferenz angenommen werden können. Bei der Implementierung fällt auf, dass in den Daten der Charité Power-Line-Interferenz bereits bei 46 Hz beginnt, wodurch diese Frequenz ebenfalls in das Dictionary zur Schätzung der Artefakte mit aufgenommen wird.

```
#Convex l1 Optimization
p = cp.Variable(360)
objective = cp.Minimize((cp.sum_squares((phi@p) - y_noise))+ 0.1*cp.norm(p, 1))
prob = cp.Problem(objective)
result = prob.solve()
p_esteem = p.value
#Estimation of PLI and BW
y_bw = phi_bw.dot(p_esteem[0:40])
y_pli = phi_pli.dot(p_esteem[40:360])
```

Listing 4.2: Schätzung der Sparse-Koeffizienten mithilfe der konvexen l_1-Norm-Optimierung.

Über die Multiplikation der geschätzten Sparse-Koeffizienten mit dem Dictionary können die beiden Artefakt-Komponenten berechnet werden und im letzten Schritt von dem verrauschten Signal abgezogen werden. Ein Beispiel für eine entrauschte Zeitreihe über die Sparse Signal Decomposition ist durch Abbildung 4.6 gegeben.

Der Quellcode zu allen drei Methoden ist im Anhang Kapitel A.2 in dem Ausschnitt A.1 zu finden. Um eine Vergleichbarkeit der Implementierungsvarianten gewährleisten zu können, ist eine geeignete Wahl von Evaluationsmetriken erforderlich. Diese werden im Folgenden beschrieben.

4.3 AUFSTELLEN DER EVALUATIONSMETRIKEN

Zur Evaluation werden vier Metriken herangezogen. In der folgenden Erläuterung der Metriken wird die Originalzeitreihe ohne Artefakt mit $x[n]$

3 https://www.cvxpy.org/

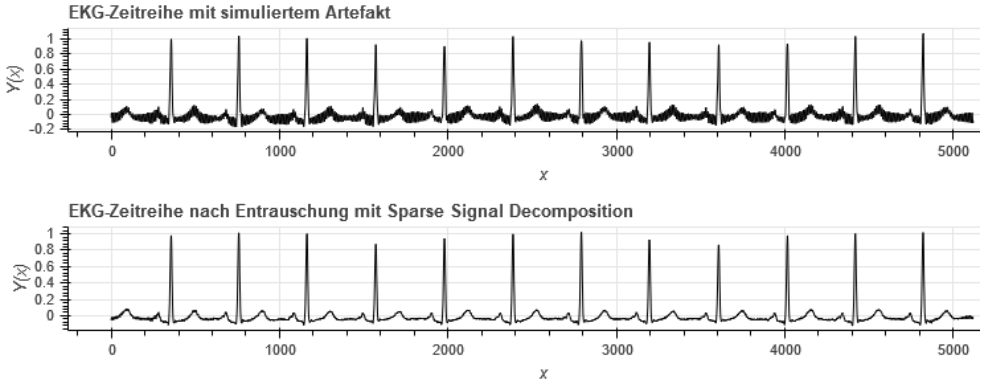

Abbildung 4.6: Oben: Zeitreihe mit simuliertem, hinzugefügtem Artefakt. Unten: Entrauschte Zeitreihe nach Anwendung der Sparse Signal Decomposition
Quelle: Eigene Darstellung

beschrieben. $\tilde{x}[n]$ gibt die entrauschte Zeitreihe an. μ_0 ist der Mittelwert der Originalzeitreihe, μ_r der entrauschten Zeitreihe. P ist die Länge der Zeitreihe.

Der Signal-to-Noise Ratio (SNR) gibt an, wie sich das Signal zu dem übergebliebenen Hintergrundrauschen verhält [23] und wird definiert durch [61]:

$$SNR = 10 \log_{10} \left[\frac{\sum_{n=1}^{P} (x[n] - \mu_0)^2}{\sum_{n=1}^{P} (x[n] - \tilde{x}[n])^2} \right] \tag{4.1}$$

Je höher daher der SNR ist, desto geringer ist der Anteil des verbleibenden Rauschens.

Der Maximum Absolute Error (MAX) gibt die maximale Abweichung zwischen der artefaktfreien Zeitreihe und der entrauschten Zeitreihe an und entspricht folgender Formel [61]:

$$MAX = max_{n=1}^{P} \{|x[n] - \tilde{x}[n]|\} \tag{4.2}$$

Diese Metrik wird verwendet, um auszuschließen, dass einzelne Punkte, wie beispielsweise der Start- und Endpunkt einer Zeitreihe, besonders schlecht geschätzt werden, dies aber aufgrund der Menge an Punkten nicht ins Gewicht fällt und daher unbemerkt bleibt.

Als dritte Metrik wird der Mean Squared Error (MSE) verwendet. Der MSE beschreibt die Summe der quadratischen Abweichungen, wodurch auch hier extreme Abweichungen mehr ins Gewicht fallen und wird definiert durch:

$$MSE = \frac{1}{P} \sum_{n=1}^{P} (x[n] - \tilde{x}[n])^2 \tag{4.3}$$

Als letztes Maß wird die Normalised Cross Correlation (NCC) genutzt. Die NCC zeigt die Korrelation zweier Zeitreihen, wobei auch Zeitreihen mit verschiedenem Wertebereich verglichen werden können. Sie bewegt sich zwischen -1 und 1. Je näher der Korrelationskoeffizient am oberen Rand des

Definitionsbereiches liegt, desto ähnlicher sind sich die Zeitreihen. Liegt der Koeffizient in der Mitte (bei Null), so ist keine Ähnlichkeit der Zeitreihen festzustellen. Bei einem Koeffizienten nahe der -1 ist ein umgekehrt proportionaler Zusammenhang gegeben. Die NCC wird definiert durch:

$$NCC = \frac{\sum_{n=1}^{P}\{(x[n] - \mu_0) \cdot (\tilde{x}[n] - \mu_r)\}}{\sqrt{\sum_{n=1}^{P}\{x[n] - \mu_0\}^2 \cdot \sum_{n=1}^{P}\{\tilde{x}[n] - \mu_r\}^2}} \tag{4.4}$$

Das nachfolgende Kapitel stellt die auf Basis der Evalustionsmetriken ermittelten Ergebnisse vor.

4.4 ERGEBNISSE

Auf Basis der zuvor vorgestellten Metriken werden die einzelnen Methoden gegenübergestellt, um die effektivste Methode als Entrauschungsprozedur auszuwählen. Bei der Implementierung der CEEMDAN-Methode hat sich gezeigt, dass diese Methode aufgrund unzureichender Geschwindigkeit nicht für die Entrauschung von großen Datenmengen geeignet ist. Trotz des Einsatztes der CEEMDAN anstelle der inperformanteren Ensemble Empirical Mode Decomposition (EEMD) dauert die Berechnung der entrauschten Zeitreihen im Testdatensatz über 13 Stunden. Die anderen Methoden benötigen im Vergleich wenige Sekunden bis wenige Minuten. Im Umfang dieser Arbeit konnte die Optimierung der CEEMDAN nicht ausgeschöpft werden, weshalb hier möglicherweise noch bessere Ergebnisse erzielt werden können. Xu et al. [73] haben gezeigt, dass der CEEMDAN in Verbindung mit einem Wavelet-Threshold-Ansatz effektiv Artefakte aus EKG-Daten entfernen kann. Da die Optimierungen der Methode in dieser Implementierung jedoch noch nicht ausgeschöpft sind, wird die CEEMDAN aus dem Vergleich mithilfe der vier Metriken ausgeschlossen. Die Ergebnisse der anderen drei Implementierungen werden durch Abbildung 4.7 visualisiert. Die genauen Zahlen werden im Anhang durch die Tabellen A.1 und A.2 inklusive der CEEMDAN dargestellt.

Die Grafik ist folgendermaßen zu interpretieren: Der SNR ist besonders gut, wenn der berechnete Wert möglichst hoch ist, da so das Verhältnis des Rauschens nach der Entrauschung im Vergleich zur Zeitreihe ohne Artefakt besonders klein ist. Der MAX-Wert ist besonders gut, wenn er möglichst klein ist, da es ein Indiz darauf ist, dass die Zeitreihen ähnlich zueinander sind. Die größte absolute Abweichung, also die größte Unähnlichkeit, ist demnach besonders klein. Der MSE ist ähnlich zu deuten. Auch hier ist ein möglichst kleiner Wert positiv, da dadurch die Summe der quadratischen Abweichungen gering ist und es daher für eine große Ähnlichkeit der Zeitreihen spricht. Als letztes ist die NCC zu benennen. Diese zeigt bei einem Wert nahe der 1 eine besonders hohe Korrelation, was in dem Falle der zu vergleichenden Zeitreihen als besonders gut aufzufassen ist. Liegt der Korrelationskoeffizient nahe 1, so ist ein hoher positiver linearer Zusammenhang zu deuten, was einer hohen Ähnlichkeit der Zeitreihen entspricht. Liegt der Koeffizient je-

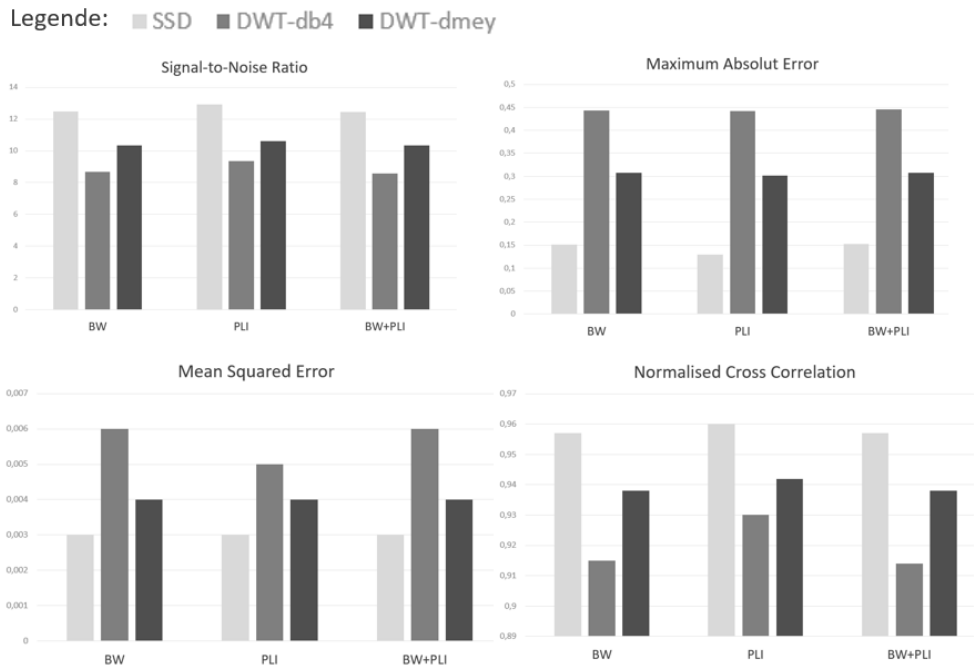

Abbildung 4.7: Die nach Artefakt aggregierten Evaluationsergebnisse, getrennt nach den drei genutzten Methoden.
Quelle: Eigene Darstellung

doch bei 0, liegt kein linearer Zusammenhang der beiden Zeitreihen vor und das Rauschen konnte demnach nicht von der Zeitreihe extrahiert werden.

Die DWT zeigt gute Ergebnisse in allen vier Metriken. Das Discrete Meyer Wavelet setzt dabei von dem Dauberchies Wavelet der Ordnung 4 ab, sodass das Discrete Meyer Wavelet bevorzugt wird. Dies entspricht den Erkenntnissen von Nguyen et al. [50]. Noch besser in Bezug auf die vier Metriken ist die Sparse Signal Decomposition einzuschätzen (in Abbildung 4.7 als SSD bezeichnet). Sowohl zur Entfernung des Baseline Wanders als auch zur Entfernung der Power-Line-Interferenz zeigt die Methode einen hohen Korrelationskoeffizienten, einen hohen SNR und geringe Abweichungen aus Basis des MAX und MSE. Auch die Kombination der beiden Artefakte lassen sich über die Sparse Signal Decomposition effektiv entfernen.

Im nächsten Schritt wird die Power Spectral Density (PSD) betrachtet, welche einen Überblick über die Frequenzspektren der Zeitreihen gibt. Die PSD ist in Abbildung 4.8 dargestellt.

Aus der Abbildung 4.8 wird ersichtlich, dass das Leistungsspektrum der durch die Sparse Signal Decomposition entrauschten Zeitreihe sowohl dem Leistungsspektrum der Originalzeitreihe als auch der Zeitreihe mit simulierten Artefakten entspricht. Die anderen drei Methoden zeigen ein deutlich geringeres Frequenzspektrum. Inbesondere die entrauschten Zeitreihen der CEEMDAN und der DWT mit dem dmey-Wavelet heben sich von den anderen Zeitreihen ab. Dies ist dadurch zu erklären, dass bei der Sparse Signal Decomposition eine deutlich geringere Frequenzspanne rausgefiltert wird, da nur die zum Baseline Wander und zur Power-Line-Interferenz gehörenden

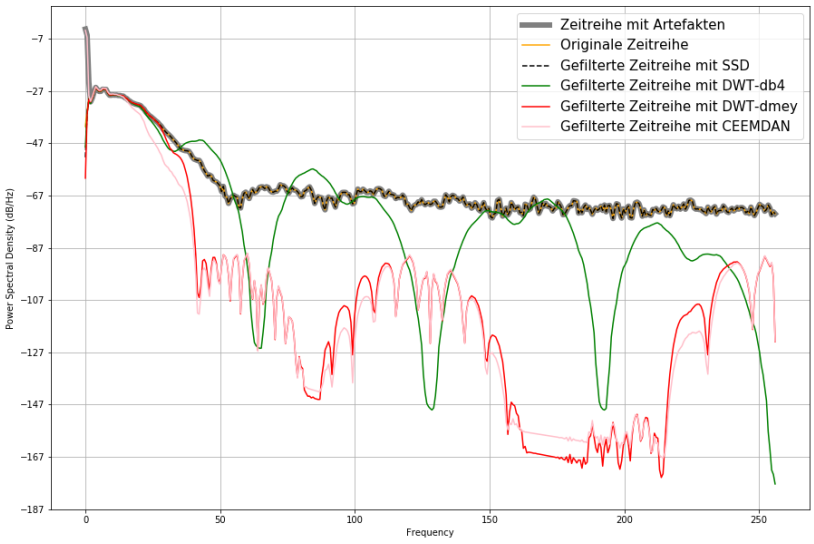

Abbildung 4.8: Power Spectral Density der verschiedenen entrauschten Zeitreihen und der Originalzeitreihe ohne Artefakt. Quelle: Eigene Darstellung mithilfe des Python-Pakets Matplotlib [44]

Frequenzen eliminiert werden. Zusätzlich wird bei der Methode eine Schätzung der Noise-Komponenten als Vorgehen gewählt, was schnell zu kleinen Unsicherheiten führt, was wiederum zu einem ausgeprägten Frequenzspektrum führt. Die erfolgreiche Eliminierung einzelner Frequenzen ist dagegen in den anderen Methoden zu beobachten.

Aufgrund der Beobachtung desselben Frequenzspektrums der Sparse Signal Decomposition und der verrauschten Zeitreihe ist herauszufinden, ob die simulierten Artefakte zufällig zu dem Entrauschungsmechanismus der Sparse Signal Decomposition passen. Daher werden die Zeitreihen zum einen für die simulierten Artefakte und zum anderen für ein beobachtetes Artefakt übereinander abgebildet und daraufhin visuell untersucht. Abbildung 4.9 zeigt in grauer Farbe ein echtes in den Daten auffindbares Artefakt. In weiteren Farben sind die verschiedenen entrauschten Zeitreihen über die verschiedenen Methoden abgebildet. Die obere Visualisierung zeigt einen Abschnitt von 10 Sekunden. Die untere entspricht einer Vergrößerung der Abbildung darüber, zur besseren Unterscheidung der angewendeten Methoden.

In der Abbildung 4.9 ist zu erkennen, dass die Sparse Signal Decomposition noch hochfrequentes Rauschen abbildet, da bei der Anwendung nur die zur Power-Line-Interferenz und zum Baseline Wander gehörigen Frequenzen rausgefiltert werden. Die weiteren Methoden zeigen deutlich geglättetere entrauschte Zeitreihen. Alle Methoden bilden für das menschliche Auge jedoch ziemlich exakt dieselbe Zeitreihe ab. Die Unterschiede sind erst durch die Vergrößerung eines Bildausschnitts zu erkennen. Dennoch ist der CEEMDAN-Algorithmus von den Zeitreihen am weitesten entfernt.

Zur weiteren Evaluation wird zusätzlich eine Zeitreihe mit simuliertem Artefakt untersucht, um herausfinden, ob das Artefakt besonders passend

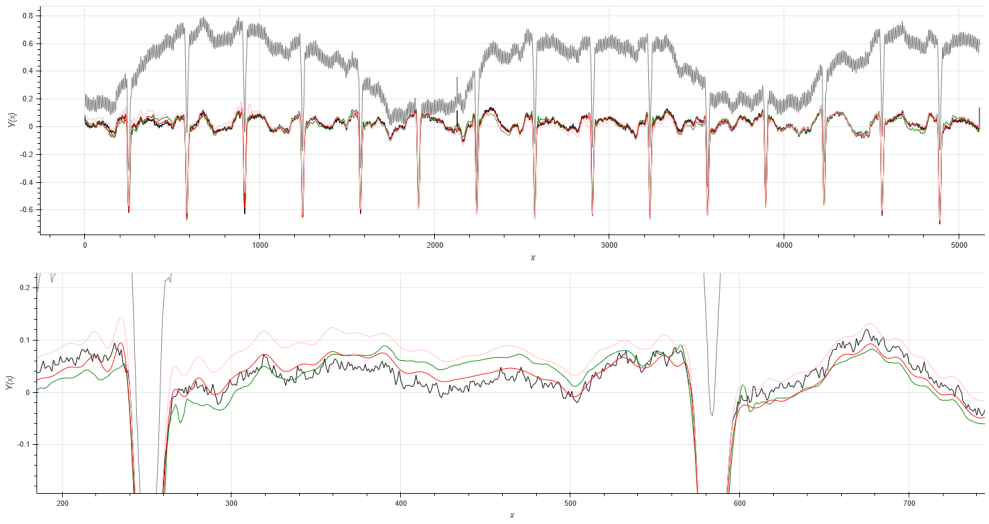

Abbildung 4.9: EKG-Zeitreihe mit beobachtetem Artefakt im Vergleich mit den eingesetzten Entrauschungsmethoden. Oben: Zeitreihe über 10 Sekunden. Unten: Zoom in die obige Zeitreihe. Legende: Zeitreihe mit Artefakt (Grau); Gefiltert mit SSD (Schwarz); Gefiltert mit DWT-db4 (Grün); Gefiltert mit DWT-dmey (Rot); Gefiltert mit CEEMDAN (Pink); Quelle: Eigene Darstellung

zur Sparse Signal Decomposition gebildet wird. Diese Zeitreihe ist in Abbildung 4.10 dargestellt.

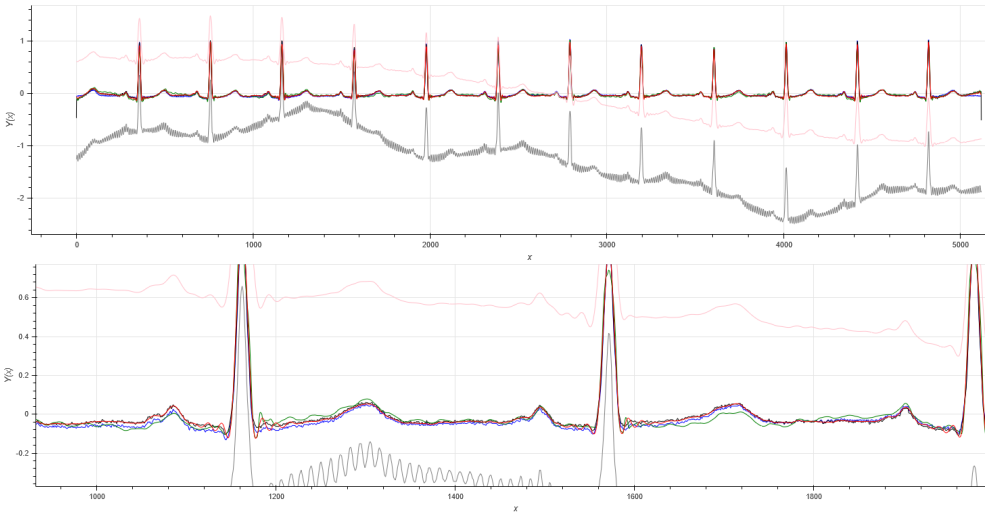

Abbildung 4.10: EKG-Zeitreihe mit simuliertem Artefakt im Vergleich mit den eingesetzten Entrauschungsmethoden. Oben: Zeitreihe über 10 Sekunden. Unten: Zoom in die obige Zeitreihe. Legende: Zeitreihe mit Artefakt (Grau); Zeitreihe ohne Artefakt (Blau); Gefiltert mit SSD (Schwarz); Gefiltert mit DWT-db4 (Grün); Gefiltert mit DWT-dmey (Rot); Gefiltert mit CEEMDAN (Pink)
Quelle: Eigene Darstellung

Die Abbildung 4.10 zeigt deutlich stärkere Abweichungen der CEEMDAN-Zeitreihe. Das Basline Wander Artefakt konnte hier nicht vollständig elimi-

niert werden. Zusätzlich fällt auf, dass die Methoden der Wavelets an den starken Frequenzänderungen, wie es beim QRS-Komplex der Fall ist, größere Ungenauigkeiten aufweisen. Der Bereich der P-Welle und T-Welle kann sehr gut rekonstruiert werden. Zudem sind die Zeitreihen deutlich glatter als die der Sparse Signal Decomposition. Ein weiterer negativer Effekt bei der Sparse Signal Decomposition ist die schlechte Schätzung an den Rändern der Zeitreihe. Insgesamt betrachtet stellen sowohl die Sparse Signal Decomposition als auch die Wavelet-Methoden mit dem dmey-Mother-Wavelet sehr gute Entrauschungsmethoden dar, welche sich nur schwer differenzieren lassen.

Für die weitere Arbeit wird die Sparse Signal Decomposition als Entrauschungsprozedur vor der Anomalieerkennung ausgewählt, da diese Methode weniger Unreinheiten bei starken Frequenzänderungen aufweist. Zudem zeigen die Metriken, dass diese Methode die Zeitreihen am besten auf Basis der generierten Artefakte entrauscht. Zur Elimination der unregelmäßigen Peaks, welche bei der Vergrößerung in die Zeitreihe aufgefallen sind, werden durch einen Moving-Average-Filter geglättet.

5

KLASSIFIKATIONSMODELL ZUR ERKENNUNG MEDIZINISCHER ANOMALIEN

In diesem Kapitel wird das Trainieren eines Klassifikationsmodells zur Erkennung von medizinischen Anomalien untersucht. Wie in Kapitel 2.2 beschrieben, gibt es drei verschiedene Anomalieerkennungsstrategien. Viele Publikationen setzten auf den Supervised- oder Semi-Supervised-Ansatz. Die Unsupervised-Anomalieerkennung wird eher selten zur Erkennung medizinischer Anomalien eingesetzt, sodass sich innerhalb dieser Arbeit auf die beiden erst genannten Ansätze fokussiert wird.

Da jedoch bei den Daten der Charité lediglich Label in Form von Alarmen durch das Monitoring-System vorliegen, welche einige Falschalarme enthalten, wird mithilfe der Prinzipien des Transfer Learnings auf den oft genutzten MIT-BIH-Arrhythmia-Datensatz zurückgegriffen. Dieser ist mehrfach evaluiert und enthält manuelle Annotationen zu jedem Herzschlag. Ziel ist es ein möglichst aussagekräftiges Modell auf diesen Daten zu trainieren und dieses daraufhin auf einen ausgewählten Datensatz der Charité-Daten zu übertragen. Dadurch wird untersucht, inwiefern Anomalien auch auf Daten aus anderen Quellen erkannt werden können und insbesondere jene Anomalien, die nicht in dem Datensatz der MIT-BIH-Datenbank berücksichtigt werden, erkannt werden können.

5.1 MODELLERSTELLUNG AUF DEM MIT-BIH-ARRHYTHMIA-DATENSATZ

In diesem Kapitel wird dafür die Konzeption des Modells auf dem MIT-BIH-Arrhythmia-Datensatz beleuchtet.

5.1.1 *MIT-BIH-Arrhythmia-Datensatz*

Die MIT-BIH-Arrhythmia-Datenbank [40] hat sich als Standard-Testdatensatz zur Entwicklung von Anomalie-Detektoren auf EKG-Zeitreihen etabliert [48, 74]. Der Datensatz enthält 48 halbstündige, ambulante EKG-Signale mit zwei Ableitungen (bezeichnet als Ableitung A und Ableitung B) von 47 Probanden. Eines dieser EKG-Signale beschreibt daher die Herzaktivität eines Patienten über den oben genannten Zeitraum, sodass mehrere Herzschläge in einer Zeitreihe vorliegen. Pro Sekunde werden 360 Punkte dargestellt, was im Folgenden als eine Aufnahmefrequenz von 360 Hz bezeichnet wird. Zu jeder Aufnahme existieren zwei Dateien. Eine Datei enthält die Daten der Aufnahme, die andere Datei ist eine Textdatei mit den Annotationen, welche jeweils zu einem Index der Zeitreihen zugeordnet werden. 23 der 48 Aufnahmen wurden vom Boston's Beth Israel Hospital (BIH) und dem Massachusetts Institute of Technology (MIT) ausgewählt, da sie einem nor-

malen Sinus-Rythmus entsprechen und daher einen Satz an normalen Herzschlägen repräsentieren. Die übrigen 25 Aufnahmen entsprechen weniger häufigen, aber klinisch signifikanten Herzanomalien. Ein Beispiel für einen Ausschnitt einer solchen Zeitreihe ist durch Abbildung 5.1 gegeben.

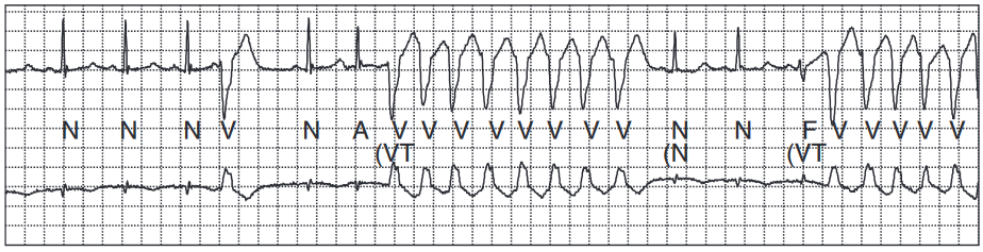

Abbildung 5.1: Zehn Sekunden des Datensatzes 205 der MIT-BIH-Arrhythmia-Datenbank inklusive der zugehörigen Herzschlag-Annotation (A: atrial premature beat, F: ventricular fusion beat, N:normal beat, V: ventricular premature beat). Obere Ableitung: II; untere Ableitung: V1
Quelle: Moody and Mark, 2001 [48]

In 45 Fällen beschreibt die Ableitung A die Extremitäten-Ableitung II. Als Ableitung B wird im Normalfall die Ableitung V1 nach Wilson gesetzt. Gelegentlich liegt anstelle der V1-Ableitung auch die V2- oder V5-Ableitung vor. In einem Fall wird die V4-Ableitung aufgezeichnet. In den anderen drei Aufzeichnungen liegt als Ableitung A die V5-Ableitung vor und für die Ableitung B in zwei Instanzen die V2 und in einem Fall die Extremitätenableitung II. Bei den Probanden handelt es sich um 25 männliche Personen im Alter zwischen 32 und 89 und um 22 weibliche Personen im Alter von 23 bis 89.

Insgesamt liegen 16 verschiedene Annotationen vor, nach denen die einzelnen Herzschläge klassifiziert werden. Die Tabelle 5.1 gibt einen Überblick über die einzelnen Herzschlag-Typen, die zugehörige Annotation und die vorkommende Häufigkeit.

Eine genaue Beschreibung des Datensatzes ist der zugehörigen Publikation „The impact of the MIT-BIH-Arrhythmia-Datenbank“ von G.B. Moody und R.G. Mark [48] zu entnehmen.

5.1.2 *Elimination von technischen Artefakten*

Auf Basis der Untersuchungen aus Kapitel 4 haben sich die Methoden der DWT mit dem dmey-Wavelet als Mother-Wavelet und der Sparse Signal Decomposition als sehr gute Entrauschungsprozeduren von EKG-Daten bewiesen. Aufgrund der besseren Elimination technischer Artefakte an den Stellen starker Frequenzänderungen kommt in der ersten Iteration die Sparse Signal Decomposition in Kombination mit einem Moving-Average-Filter zum Einsatz.

Zur Implementierung der Bereinigung der Daten werden die 48 Aufnahmen Schritt für Schritt eingelesen und mithilfe der Sparse Signal Decompo-

Herzschlag-Typ	Annotation	Anzahl
Normal Beat (NOR)	N	75017
Left Bundle Branch Block (LBBB)	L	8072
Right Bundle Branch Block (RBBB)	R	7255
Artrial Premature Contraction (APC)	A	2546
Premature Ventricular Contraction (PVC)	V	7129
Paced Beat (PACE)	\	7024
Aberrated Atrial Premature Beat (AP)	a	150
Ventricular Flutter Wave (VF)	!	472
Fusion of Ventricular and Normal Beat (VFN)	F	802
Blocked Atrial Premature Beat (BAP)	x	193
Nodal (Junctional) Escape Beat (NE)	j	229
Fusion of Paced and Normal Beat (FPN)	f	982
Ventricular Escape Beat (VE)	E	106
Nodal (Junctional) Premature Beat (NP)	J	83
Atrial Escape Beat (AE)	e	16
Unclassifiable Beat (UN)	Q	33
Summe	**16**	**110109**

Tabelle 5.1: Anzahl und Annotationsbeschreibung der einzelnen Herzschlag-Typen der MIT-BIH-Arrhythmia-Datenbank
Quelle: Ye et al., 2012 [74] und Webseite der PhysioBank-Annotationen [52]

sition das Baseline Wander und die Power-Line-Interfernz eliminiert sowie die Zeitreihen im Nachhinein geglättet. Da die hier verwendete Datenmenge deutlich höher ist als die Datenmenge, die zuvor für die Evaluierung der Methode eingesetzt wurde, zeigt sich erst im Durchlauf des Bereinigungsprozesses, dass die Methode bei einer solchen Datenmenge einen deutlich höheren Rechenaufwand aufweist. Die Dauer des ersten Durchlaufs beträgt 75 Stunden. Um eine schnellere Reproduzierbarkeit in Zukunft zu gewährleisten, wird ab der zweiten Iteration die Methode der DWT eingesetzt, die ebenfalls gute Ergebnisse lieferte. Diese zeigt auch bei einer hohen Datenmenge eine hohe Geschwindigkeit.

5.1.3 *Herzschlagerkennung und -segmentierung*

Zur Herzschlagerkennung wird der R-Peak-Erkennungsalgorithmus von Elgendi et al. [18] eingesetzt, welcher mithilfe von zwei Moving-Average-Fenstern die Lokation der R-Peaks erkennt. Ein Beispiel der R-Peak-Erkennung ist durch die Zeitreihe in Abbildung 5.2 dargestellt.

Die roten Punkte visualisieren die vom Algorithmus gefundenen Punkte der R-Peaks. Da es, wie bereits in Kapitel 2.4 beschrieben, eine Schwierigkeit darstellt, bei Anomalien die T- und P-Welle zu erkennen, wird ein fes-

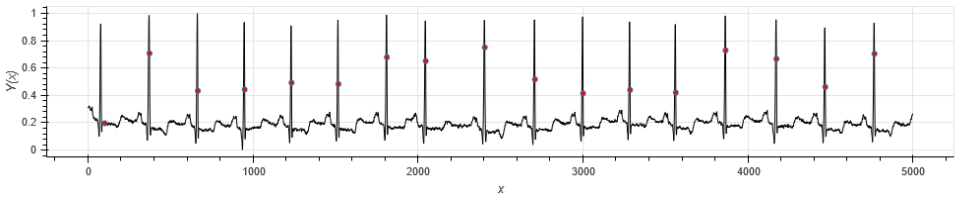

Abbildung 5.2: Visualisierung des Ergebnisses des Two-Moving-Average-Algorithmus nach Elgendi et al. [18].
Quelle: Eigene Darstellung

tes Fenster definiert, welches jeden Herzschlag basierend auf dem erkannten R-Peak gleichermaßen extrahiert. Die Annotationsdateien der MIT-BIH-Datenbank geben zwar auch einen Punkt jedes Herzschlags mit, dieser variiert jedoch leicht in der Position. Zudem ist es schwierig zu gewährleisten die identische Stelle in dem Datensatz der Charité zu finden, da dieser über keine manuelle Annotationsdatei verfügt. Daher ist das Ziel, beide Datensätze auf dieselbe Art und Weise vorzuverarbeiten, sodass die Klassifikation möglichst gut gelingt. Ausgehend vom R-Peak wird das Fenster so gesetzt, dass entsprechend dem Paper von Ye et al. [74] 100 Punkte nach links und 200 Punkte nach rechts extrahiert werden. Diese Größe wird gewählt, da eine Frequenz von 360 Hz vorliegt und das Herz pro Sekunde in der Regel zwischen einem und zwei Mal schlägt. Ein Beispiel für einen extrahierten Herzschlag zeigt die Visualisierung 5.3.

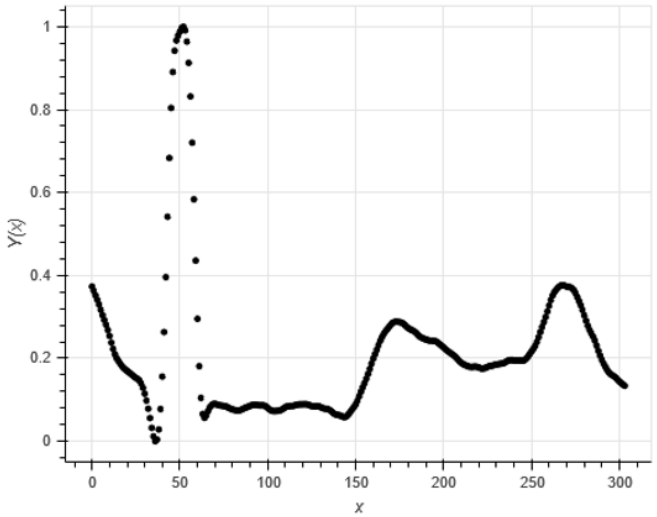

Abbildung 5.3: Visualisierung eines extrahierten Herzschlag-Segments
Quelle: Eigene Darstellung

Vor der Extraktion wird geprüft, ob sich eine Annotation in der direkten Umgebung von plus/minus 30 Punkten befindet, sodass das Herzschlagsegment mit einer Annotation versehen werden kann. Ein Filter auf die Annotationen führt dazu, dass nur die Herzschläge extrahiert werden, die über eine Annotation bezüglich des Herzschlagtyps verfügen. Eine Übersicht über die

Anzahl der extrahierten Herzschläge pro Typ ist aus Tabelle 5.2 zu entnehmen.

Annotation	Anzahl	Annotation	Anzahl
N	9726	j	197
R	7136	!	136
/	5410	a	85
V	5271	J	83
L	5112	E	82
A	2539	x	64
f	913	Q	25
F	679	e	16

Tabelle 5.2: Anzahl der extrahierten Herzschläge pro Annotation

Wie zu erwarten ist zu sehen, dass die Daten unbalanciert sind. Die normalen Herzschläge (Annotation: N) zeigen die größte Anzahl. Die weiteren Annotationen beziehen sich auf Anomalien, weshalb sie dementsprechend weniger oft auftreten. Der Datensatz wurde vom MIT und BIH bereits so ausgewählt, möglichst viele Anomalien zu zeigen, jedoch sind einige dennoch sehr selten. Die größte Anzahl ist bei einem Right Bundle Branch Block (Annotation: R), Paced Beat (Annotation: /), Premature Ventricular Contraction (Annotation: V), Left Bundle Branch Block (Annotation: L) und einer Artrial Premature Contraction (Annotation: A) zu erkennen. Da Anomalien, wie der Name schon sagt, in der Regel nicht häufig zu beobachtende Ereignisse sind, ist der Umgang mit unbalancierten Daten ein typisches Problem der Anomalieerkennung, welches im Zuge der Klassifikation auch bei diesen Daten berücksichtigt werden muss.

5.1.4 *Feature-Extraktion*

Die Feature-Extraktion wird in Anlehnung an das Paper von Ye et al. [74] durchgeführt, da diese mit einem traditionellen Machine-Learning-Ansatz eine sehr gute Modell-Performance in Bezug auf die oft genutzte *Accuracy*-Metrik auf den MIT-BIH-Daten bei der Anomalieerkennung erreichten. Li et al. [37] stellten einige Ansätze zur Herzschlagklassifikation in einer Tabelle gegenüber. Diese Übersicht ist dem Anhang im Kapitel B.1 angefügt. Der Fokus auf die traditionellen Machine-Learning-Verfahren hat den Hintergrund, dass das Cluster, auf dem die Auswertungen implementiert werden, über keine GPU verfügt, weshalb die weniger rechnenintensiven Varianten bevorzugt werden.

Wie in den theoretischen Grundlagen (Kapitel 2.5) erläutert, besteht die Feature-Extraktion aus den zwei Schritten Feature-Konstruktion und -Selektion. Die Feature Konstruktion basiert auf dem Erstellen neuer Feature auf Grundlage der vorliegenden Daten oder darauf bestehende Merkmale als Feature zu betrachten. Zur Klassifikation von Herzschlägen wird auf den

ersten Ansatz zurückgegriffen, sodass neue Feature auf Basis der Zeitreihenpunkte des Herzschlags erstellt werden. Dabei werden zum einen morphologische Feature extrahiert, welche eine Aussage über die Gestalt des Herzschlags machen können und zum anderen dynamische Feature, welche das dynamische Zusammenspiel mehrerer Herzschläge betrachten. Die Berechnung dieser Feature müssen jedoch bereits vor der Herzschlagsegmentierung erfolgen, da nur vor der Segmentierung noch Informationen zu den nebenliegenden Herzschlägen vorliegen. Als morphologische Feature werden Hauptkomponenten, die auf Basis von ausgewählten DWT-Koeffizienten und unabhängigen Komponenten erstellt werden, genutzt. Die Definition der dynamischen Feature erfolgt über RR-Intervalle. Die weiteren Unterkapitel gehen auf die einzelnen Feature genauer ein.

Diskrete Wavelet-Transformation

Im ersten Schritt der Feature-Konstruktion werden die DWT-Koeffizienten extrahiert, welche einen hohen Informationsgehalt in Bezug auf die EKG-Komponenten besitzen [62]. Dafür wird das Dauberchies Wavelet der Ordnung 8 eingesetzt und die Detailkoeffizienten D3, D4 und der Approximationskoffizient A4 ermittelt. Sie treffen bei dem 360 Hz-Signal eine Aussage über die Frequenzbereiche 22,5-45 Hz, 11,45-22,5 Hz und 0-11,45 Hz [50]. Insgesamt werden 114 Koeffizienten für jeden Herzschlag auf diese Weise berechnet (32 aus D3, 32 aus D4 und 50 aus A4 [74]).

Unabhängigkeitsanalyse

Zusätzlich zu den DWT-Koeffizienten helfen die aus einer Unabhängigkeitsanalyse resultierenden unabhängigen Komponenten die Morphologie eines Herzschlags zu definieren. Sie werden aus den Datenpunkten des vorliegenden Herzschlags gebildet. Ursprünglich diente die Unabhängigkeitsanalyse zur Trennung unabhängiger Signalkomponenten aus einer Reihe beobachteter Signale, wobei nur wenige Vorabinformationen vorliegen [74]. In diesem Zusammenhang liegt der Nutzen der Unabhängigkeitsanalyse darauf, unabhängige Komponenten eines EKG-Signals als neue Feature zu konstruieren, da sie wichtige Hinweise auf die Morphologie des Herzschlags geben können. Ye et al. untersuchten mit Hilfe einer zehnfachen Kreuzvalidierung, wie viele Komponenten in einem Bereich von 10 bis 30 Komponenten die besten Ergebnisse bei der Klassifikation bringen [74]. Sie entschieden sich für 14 Komponenten, weshalb auch hier 14 unabhängige Komponenten für jeden Herzschlag extrahiert werden.

Hauptkomponentenanalyse

Mithilfe der DWT und der Unabhängigkeitsanalyse können für jeden Herzschlag 128 Feature (114 DWT-Koeffizienten und 14 ICA-Feature) extrahiert werden. Da es sich um eine große Menge an Feature handelt, wird zur Dimensionsreduktion eine Hauptkomponentenanalyse durchgeführt. Zur Entscheidung bezüglich der Anzahl der Hauptkomponenten, wird eine zehnfa-

che Kreuzvalidierung in einer Vorwärtsselektion durchgeführt. Zur zehnfachen Kreuzvalidierung wird mithilfe einer Support Vector Machine ein Modell zur Vorhersage der sechs am stärksten vertretenen Anomalien berechnet, wobei nur die erste Hauptkomponente als Feature mit hinein gegeben und die durchschnittliche Accuracy berechnet wird. In jedem weiteren Schritt wird die nachfolgende Hauptkomponente mit hinzugegeben, bis eine Anzahl von 30 Hauptkomponenten erreicht ist, was dem Vorgehen einer Vorwärtsselektion entspricht. Das Ergebnis der zehnfachen Kreuzvalidierung pro Modell ist durch Abbildung 5.4 visualisiert.

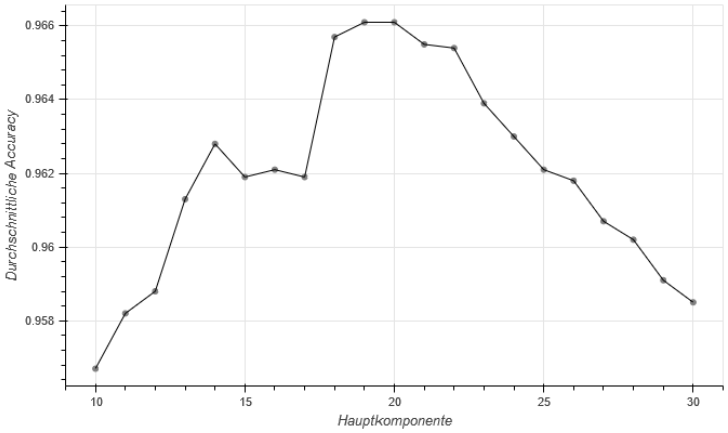

Abbildung 5.4: Zehnfache Kreuzvalidierung zur Bestimmung der Anzahl an Hauptkomponenten
Quelle: Eigene Darstellung

Aus der Grafik 5.4 geht hervor, dass die größte Accuracy bei 19 bzw. 20 Hauptkomponenten erreicht wird. Daher wird die Anzahl von 19 Hauptkomponenten für die Definition der morphologischen Feature festgesetzt.

RR-Intervalle

Wie zuvor erläutert, ist die Definition der RR-Intervalle bereits vor der Segmentierung in einzelne Herzschläge notwendig. Da sie jedoch zu dem Themenbereich der Feature-Extraktion gehören, werden sie in diesem Kapitel aufgeführt.

Die Abstände zwischen den aufeinander folgenden QRS-Komplexen treffen ebenfalls eine relevante Aussage über einen normalen EKG-Verlauf. Daher werden vier verschiedene RR-Intervalle aus den Daten ermittelt und als dynamische Feature definiert. Das *previous RR-Intervall* gibt den Abstand des aktuellen R-Peaks zum R-Peak des vorherigen Herzschlags an. Das *post RR-Intervall* beschreibt den Abstand vom aktuellen R-Peak zum nachfolgenden R-Peak. Zur Berücksichtigung des Trends werden zusätzlich durchschnittliche Abstände ermittelt. Zum einen beschreibt das *local RR-Intervall* die durchschnittlichen Abstände zwischen den R-Peaks der letzten zehn Sekunden und zum anderen charakterisiert das *average RR-Intervall* die Abstände zwischen den R-Peaks der letzten vier Minuten.

Nach der Extraktion der RR-Intervalle stehen insgesamt 19 morphologische Feature und vier dynamische Feature zur Klassifikation durch ein Modell bereit. Ye et al. [74] extrahierten diese 19 Feature für beide beobachteten Ableitungen, sodass zwei unabhängige Klassifikatoren erstellt und verglichen werden konnten. Da bei den Daten der Charité bisher jedoch nur die Extremitätenableitungen bereitstehen und die zweite Ableitung in den MIT-BIH-Daten von Ableitungen nach Wilson geprägt ist, ist das Ziel lediglich einen Klassifikator auf Basis der Ableitung II zu konzipieren. Daher werden die 23 Feature ausschließlich für die Ableitung A extrahiert.

5.1.5 *Klassifikation*

Die Herzschlagklassifikation dient der Vorhersage von abnormalen und normalen Herzschlägen. Aufgrund der guten Ergebnisse von Ye et al. [74] wird eine SVM als Klassifizierer eingesetzt. Zum Vergleich werden zusätzlich zwei weitere traditionelle Machine-Learning-Klassifizierer implementiert, wobei die Wahl auf einen KNN-Klassifizierer und die LDA fällt, da sie ebenfalls aus dem Bereich der traditionellen Machine-Learning-Verfahren erfolgreich eingesetzt wurden. Da jedoch die SVM auch im Vergleich die beste Accuracy, Sensitivität und die geringsten Falsch-Alarme zeigt, wird im Weiteren lediglich auf die SVM eingegangen. Alle drei Metriken sind relevant, da es sich um unbalancierte Daten handelt. Die Ergebnisse der beiden anderen Algorithmen sind im Anhang im Kapitel B (Tabelle B.1 und B.2) zu finden.

Zur Verbesserung der Modell-Performance der SVM wird ein *Grid Search* als Hyperparameteroptimierung über einige Variablen durchgeführt. Bei einem Grid Search wird für alle Kombinationen der angegebenen Hyperparameter ein Modell erstellt und im Anschluss verglichen, sodass die optimale Kombination festgestellt werden kann. Die Bereiche, über die das beste Modell gesucht wird, werden in Tabelle 5.3 dargestellt.

Die SVM-Implementierung von scikit-learn[1] besitzt einen zusätzlichen Parameter *class_weight*. Dieser dient bei unbalancierten Daten zur unterschiedlichen Klassengewichtung. Ist er nicht gesetzt, so wird jede Klasse mit eins gewichtet. Da in diesem Falle unbalancierte Daten vorliegen, wird der Parameter als 'balanced' mitgegeben, sodass die Klassengewichte im SVM-Algorithmus angepasst werden. Das finale Modell, welches die beste Accuracy aufweist, setzt sich aus folgenden Parametern zusammen: C=10, Kernel='rbf', Gamma='scale, Shrinking='True'. Der Parameter Degree ist für den RBF-Kernel irrelevant, da er dem Grad des polynomialen Kernels entspricht.

Auf Basis dessen werden zwei Modelle trainiert. Zum einen wird ein Modell zur Klassifikation der sechs Herzschlagtypen, die die höchste Anzahl an Vorkommnissen aufweisen (N, R, /, V, L, A), erstellt. Dieses Modell ist insbesondere zum Vergleich mit den Ergebnissen anderer Paper gedacht. Außerdem dient es der expliziten Erkennung spezieller Ausprägungen einer Anomalie. Da ausreichend Trainingsdaten zum Trainieren des Modells vor-

1 https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Parameter	Beschreibung	Bereich
C	C ist ein Regularisierungsparameter, wobei die Regularisierung umgekehrt proportional zu C implementiert ist. Es handelt sich um eine L2-Regularisierung.	[1-20]
Kernel	Spezifiziert, welcher Kernel genutzt wird.	['linear', 'poly', 'rbf', 'sigmoid']
Degree	Spezifiziert den Grad der polynomialen Kernel-Funktion. Bei allen anderen Kernels wird der Grad ignoriert.	[2-5]
Gamma	Definiert welcher Koeffizient für die Kernels rbf, poly und sigmoid genutzt werden. 'scale' bedeutet: gamma = 1 / (Featureanzahl * Varianz von X). 'auto' bedeutet: gamma = 1 / Featureanzahl	['scale', 'auto', 0.1, 0.2]
Shrinking	Sagt aus, ob eine Shrinking-Heuristik verwendet wird.	[True, False]

Tabelle 5.3: Übersicht über die Parameter, welche beim Grid Search über die angegebenen Bereiche optimiert werden.

liegen müssen, werden alle Herzschlagtypen miteinbezogen, welche mehr als 2000 Beobachtungen aufweisen.

Zum anderen wird ein binäres Modell entwickelt, welches lediglich nach Anomalie und Normal klassifiziert. Dieses setzt sich in der Klasse Anomalie jedoch auf Basis aller in dem MIT-BIH-Datensatz vorkommenden 15 Anomalien zusammen. Es ist speziell auf den Anwendungsfall der Charité angepasst, da das Monitoring-System der Charité andere Alarme als die klassifizierten Anomalien ausgeben, sodass eine Abweichung von einem normalen Herzschlag als Anomalie klassifiziert werden soll. Dort sollen die Stellen markiert werden, die für die medizininschen Forscher*innen als relevant eingestuft werden.

Ye et al. [74] implementierten zusätzlich zu dem hier entwickelten Klassifikator für Ableitung A auch einen für Ableitung B. Zur finalen Vorhersage nutzten sie zwei Vorgehen. Der *Rejection-Ansatz*, bei dem die Herzschläge, die von beiden Modellen unterschiedlich klassifiziert wurden, zur weiteren manuellen Überprüfung keine Vorhersage bekamen. Die zweite Variante entsprach dem *Bayesianischen-Ansatz*, welcher die Wahrscheinlichkeitsschätzungen der beiden einzelnen Klassifikatoren berücksichtigt, um eine finale Entscheidung zu treffen. Beide Methoden erhöhten die Genauigkeit noch ein wenig. Dieser Ansatz ist aus dem Grund, dass bei den Daten der Charité derzeit keine Ableitungen nach Wilson vorliegen, nicht umsetzbar.

5.2 ANWENDUNG AUF DEN CHARITÉ-DATENSATZ

Basierend auf den Prinzipien des Transfer Learning wird das auf den Daten der MIT-BIH Arrhythmia Database trainierte Modell auf einer ausgewählten Untermenge des Charité-Datensatzes ausgeführt. Die Auswahl wird mithilfe der Philips-Monitoring-Annotationen getroffen. Es werden sowohl abnormale Herzschläge als auch normale Herzschläge miteinbezogen. Die abnormalen Herzschläge entsprechen annotierten Daten der Klassen Bradykardie, Bigeminus, ST-Streckenerhebung und Vorhofflimmern.

Nach der Zusammenstellung des Charité-Testdatensatzes werden diese Daten auf dieselbe Weise vorverarbeitet, wie die Daten der MIT-BIH-Datenbank. Da es sich bei den Charité-Daten um 512-Hz-Aufnahmen handelt, müssen die Daten zunächst ein Downsampling zum Mapping auf eine 360-Hz-Aufnahme durchlaufen, um den Feature dieselbe Aussagekraft zu geben. Der Ablauf der einzelnen Preprocessing-Schritte ist durch die Abbildung 5.5 gegeben. Als Downsampling wird ein Fenster verwendet, welches alle 512 Datenpunkte 360 aus der im Fenster befindenen Grundmenge von 512 Datenpunkten zufällig zieht. Es wird von einer statistischen Gleichverteilung ausgegangen.

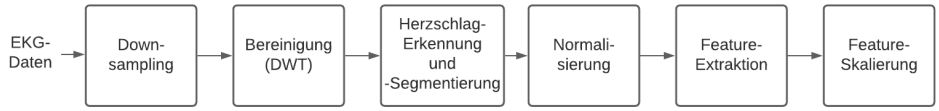

Abbildung 5.5: Ablaufplan der Vorverarbeitung der Charité-Daten
Quelle: Eigene Darstellung

Insgesamt werden auf diese Weise 140 Herzschläge extrahiert und daraufhin mit einem Label versehen. Da es sich um andere Anomalieklassen als bei dem MIT-BIH-Datensatz handelt, wird ein binäres Labeling verwendet. Alle Anomalieklassen werden zu einer Klasse zusammengefasst. Die normalen Herzschläge repräsentieren die zweite Klasse. Diese Daten werden über den binären Klassifizierer aus Kapitel 5.1.5 klassifiziert.

5.3 AUFSTELLEN DER EVALUATIONSMETRIKEN

Die Modell-Performance wird mithilfe von drei Metriken evaluiert. In den meisten Studien wird als Vergleichswert die Accurcy angeben. Da es sich um unbalancierte Daten handelt, reicht die Betrachtung der Accuracy nicht aus. Wird eine Klasse mit wenigen Beobachtungen systematisch falsch geschätzt, so wird diese Falschschätzung durch die wenigen Repräsentationen nur durch einen geringen Anteil in der Accuracy abgebildet. Als weitere Metriken werden daher die Sensitivität und der prozentuale Anteil an Falsch-

Alarmen ausgewählt. Die Sensitivität gibt an, wie oft eine Anomalie bei einer vorliegenden Anomalie tatsächlich erkannt wird:

$$\frac{TP}{TP + FN}, \tag{5.1}$$

mit TP = *True Positive* und FN = *False Negative*. Die Falsch-Alarm-Rate besagt, wie oft ein Herzschlag als Anomalie klassifiziert wird, obwohl er keiner Anomalie entspricht:

$$\frac{FP}{FP + TN}, \tag{5.2}$$

mit FP = *False Positive* und TN = *True Negative*. Die Sensitivität ist bei medizinischen Daten höher zu gewichten, da es zu einer manuellen Überprüfung durch medizinisches Personal führt. Die Tatsache, Anomalien nicht zu entdecken, aber weniger Falsch-Alarme zu haben ist schlechter zu bewerten als alle Anomalien zu entdecken, aber dafür eine höhere Anzahl gesunder Herzschläge als Anomalien zu deklarieren.

5.4 ERGEBNISSE UND DISKUSSION

Die Ergebnisse der Modelle zur Klassifikation nach „abnormal“ und „normal“ sowie zur Klassifikation der sechs meistvertretenden Herzschlagtypen der MIT-BIH-Arrhythmia-Datenbank sind der Tabelle 5.4 zu entnehmen. Die Tabelle zeigt die Ergebnisse auf dem unabhängigen Testdatensatz der MIT-BIH-Arrhythmia-Datenbank, welcher zuvor aus dem erstellten Datensatz extrahiert wurde. Es wurde ein *Stratified ShuffleSplit* angewandt, sodass aus jeder vorliegenden Klasse 20% der Daten als Testdatensatz definiert wurden. Die hier gezeigten Ergebnisse basieren auf der Klassifikation nach der Ableitung A im MIT-BIH-Arrhythmia-Datensatz, da diese hauptsächlich die EKG-Ableitungen enthält, die auch auf dem Charité-Datensatz verfügbar sind. Bei der Klassifikation auf den Merkmalen basierend auf der Ableitung B kann eine leicht bessere Accuracy, Sensitivität und ein leicht besserer prozentualer Anteil an Falsch-Alarmen vernommen werden, wie im Anhang in der Tabelle B.1 zu sehen ist.

Typen	TP	FP	TN	FN	Sensitivität	Falsch-Alarme	Accuracy
6	4.828	12	1.923	28	99,75 %	0,62 %	99,13 %
2	5.275	26	1.906	29	99,45 %	1,36 %	99,23 %

Tabelle 5.4: Ergebnisse der ausgewählten Modelle zur Klassifikation von sechs Herzschlagtypen in der oberen Zeile und einer binären Klassifikation, bei dem 16 verschiedene Typen mit einbezogen wurden, in der unteren Zeile.

Die Ergebnisse, sowohl der Klassifikation von sechs Herzschlagtypen als auch der binären Klassifikation nach „abnormal“ und „normal“ auf Basis von 16 Herzschlagtypen, zeigen auf dem MIT-BIH-Datensatz eine sehr gute Accurcy von 99,13% bzw. 99,23 %, eine hohe Sensitivität von 99,75 % bzw.

99,45 % und einen geringen prozentualen Anteil an Falsch-Alarmen von 0,62 % bzw. 1,36 %. Dies zeigt die Effektivität des Zusammenspiels der vorgestellten Methodensammlung zur Anomalieerkennung. Diese besteht aus der Bereinigung mithilfe der DWT, der Herzschlagerkennung und -segmentierung über einen Two-Moving-Average-Detektor und einem festen Fenster zur Segmentierung, der Feature-Extraktion über die DWT-Koeffizienten, der unabhängigen Komponenten, der Haptkomponentenanalyse und der RR-Intervalle sowie der Klassifikation verschiedener abnormaler und normaler Herzschläge über eine SVM. Die Ergebnisse der Klassifikation über einen KNN-Klassifikator oder eine LDA fallen mit einer Accuracy von 98,66 % und 82,14 % schlechter aus, wie im Anhang Kapitel B zu entnehmen ist.

Das binäre Modell lässt sich allerdings nicht direkt auf den Charité-Datensatz übetragen. Als Testdaten werden verschiedene Anomalien verwendet, welche nicht im MIT-BIH-Datensatz vorkommen, da das Monitoring-System der Charité bei anderen Anomalien einen Alarm auslöst. Es handelt sich bei den Daten der Charité um eine zu große Datenmenge, die nicht vollständig in das Modell überführt werden kann, da daraufhin eine manuelle Überprüfung aus medizinischer Sicht der Ergebnisse durchgeführt werden müsste. Daher wurden Anomalien verwendet, die mit dem Auge eindeutig als diese klassifiziert werden können und über das Alarm-System gefunden werden. Zudem wurden einige normale EKG-Signale ebenfalls hinzugefügt. Das Ergebnis des ersten Versuchs ist durch Tabelle 5.5 dargestellt. Auch hier zeigt die Tabelle die Ergebnisse auf einem zuvorigen extrahierten Testdatensatz, welcher 20% der Daten aus jeder Klasse enthält.

Typen	TP	FP	TN	FN	Sensitivität	Falsch-Alarme	Accuracy
2	32	93	3	14	69,57 %	96,88 %	24,65 %

Tabelle 5.5: Ergebnis der ersten Klassifikation auf den Charité-Daten. Das Experiment basiert auf derselben Datenvorbereitung wie das Modell mit der höchsten Accuracy aus Kap. 5.4.

Das erste Modell klassifiziert fälschlicherweise einen Großteil der Herzschläge als Anomalie, auch wenn eine große Anzahl normaler Herzschläge vorliegt. Es wird angenommen, dass dies an der unterschiedlichen Auflösung der Daten liegt und die DWT-Koeffizienten sehr stark auf eine Zeitverschiebung reagieren. Daher werden die Hauptkomponenten in weiteren Experimenten direkt auf Basis der downgesampleten Datenpunkte und der unabhängigen Komponenten gebildet. Zudem wird das Bereinigen über die DWT ausgelassen. Die Ergebisse des Testdatensatzes werden mithilfe der Tabelle 5.6 dargelegt.

Typen	TP	FP	TN	FN	Sensitivität	Falsch-Alarme	Accuracy
2	42	55	39	4	91,30 %	58,51 %	57,86 %

Tabelle 5.6: Ergebnis der zweiten Klassifikation auf den Charité-Daten. Das Experiment zeigt eine Verarbeitung ohne die Wavelet-Transformation.

Aus den Ergebnisse lässt sich die Hypothese ableiten, dass sowohl die Bereinigung der Rohdaten zur Vorhersage eine große Rolle spielt als auch, dass eine übereinstimmende Auflösung der Signale zum Schaffen einer gleichen Ausgangssituation relevant ist. Durch das Auslassen der DWT konnte das Modell um 33,21 % verbessert werden, jedoch kann es noch keine verlässliche Vorhersage von Anomalien auf externen Datensätzen machen. Ein manuelles Labeling auf einem repräsentativen Teil der Daten eines der verwendeten Monitoring-Systeme könnte die Vorhersage und die Falsch-Alarm-Rate der Systeme deutlich verbessern sowie die medizinischen Forscher*innen bei der Langzeitanalyse unterstützen. Es würde eine Entwicklung eines Modells direkt auf den Daten der Charité ermöglichen. Das Kapitel 5.1.5 zeigte wie gut der Supervised-Ansatz auf den Daten des MIT-BIH-Datensatzes funktioniert, weshalb der Ansatz auch direkt auf dem Zieldatensatz vielversprechend sein könnte ohne die Prinzipien des Transfer Learning nutzen zu müssen. Eine weitere Möglichkeit wäre die Verbesserung des Downsampling-Algorithmus, um so eventuell eine gleiche Datenbasis schaffen zu können. Weitere Methoden wie die Verwendung eines Autoencoders oder andere neuronale Netze wurden bisher noch nicht betrachtet. Auch diese könnten einen vielversprechenden Ansatz zur Erkennung von Anomalien beschreiben, sofern die Infrastruktur bereitsteht.

6 FAZIT UND AUSBLICK

In diesem Kapitel wird ein Fazit gezogen und anschließend ein Ausblick auf potentielle aufbauende Untersuchungsaspekte gegeben.

6.1 FAZIT

Ziel dieser Masterarbeit war die Spark-basierte Analyse von Monitoring-Daten - im Speziellen die des Parameters Elektrokardiogramm - intensiv beobachteter Patienten der Charité hinsichtlich der Anomalieerkennung. Zu diesem Zweck wurden zwei Fragestellungen analysiert:

1. Welche Methode eignet sich zur automatisierten Elimination technischer Artefakte, um die ursprünglichen realen Vitalwerte des Patienten wiederherzustellen?
2. Wie können medizinische Anomalien zuverlässig und automatisiert-mithilfe eines Klassifikationsmodells gefunden werden?

Methodenauswahl zur automatisierten Elimination technischer Artefakte

Zur Beantwortung der ersten Fragestellung wurden in Kapitel 2.3 auf Basis des aktuellen Forschungsstands drei effektive Methoden identifiziert und vorgestellt: Sparse Signal Decomposition, DWT und CEEMDAN. Bei der DWT hängt das Ergebnis stark von der richtigen Wahl der Basisfunktion ab. Daher wurden sowohl das oft eingesetzte db4-Wavelet [5, 37] und das bei Nguyen et al. [50] favorisierte dmey-Wavelet in den Vergleich der Methoden aus Kapitel 4 miteinbezogen. Hier zeigten insbesondere die Sparse Signal Decomposition und die DWT mit dem dmey-Wavelet sehr gute Ergebnisse zur Elimination der technischen Artefakte Baseline Wander und Power-Line-Interferenz. Die CEEMDAN schnitt dagegen nicht gut ab, was jedoch nicht direkt mit der Funktionsfähigkeit der Methode zusammenhängt, sondern mit dem hohen Rechenaufwand auf einer kleinen Datenmenge, weshalb diese nicht weiter optimiert wurde. Beide in Betracht gezogenen Methoden haben ihre Vor- und Nachteile:

Die Sparse Signal Decomposition wird dazu genutzt, um gezielt einzelne Frequenzen herauszufiltern. Dies führt dazu, dass nach der Bereinigung keine geglättete Funktion vorliegt und daher zusätzlich beispielsweise ein Moving-Average-Filter eingesetzt werden sollte. Die Kombination dieser beiden Methoden zeigt sehr gute Ergebnisse in Bezug auf den SNR, MAX, MSE und NCC zur Elimination von Baseline Wander und Power-Line-Interferenz. In Kapitel 5 hat sich herausgestellt, dass aufgrund des hohen Rechenaufwands die Sparse Signal Decomposition nicht für die Anwendung auf ei-

ner großen Datenmenge geeignet ist. Die DWT zeigt dagegen auch bei einer großen Datenmenge einen geringen Rechenaufwand. Auch hier können die beiden technischen Artefakte Baseline Wander und Power-Line-Interferenz effektiv bereinigt werden. Negativ zu sehen ist, dass eine höhere Verzerrung an Punkten starker Frequenzänderungen, wie beim QRS-Komplex, zu verzeichnen ist.

Insgesamt ist daher festzuhalten, dass zum praktischen Einsatz auf große Datenmengen sich die DWT besser eignet, da der Rechenaufwand deutlich geringer ist, als die der Sparse Signal Decomposition. Wird jedoch nur ein kleiner Datensatz bereinigt, so ist die Sparse Signal Decomposition eher zu empfehlen, da die technischen Artefakte besser eliminiert werden können, ohne dass Stellen mit starken Frequenzänderungen unsauber wiederhergestellt werden.

Entwicklung eines Vorgehens zum Finden medizinischer Anomalien

Zur Beantwortung der zweiten zu analysierenden Fragestellung wurde auf Basis der Prinzipien des Transfer Learnings der öffentliche MIT-BIH-Datensatz herangezogen, welcher sich als Standard-Testdatensatz zur Entwicklung von Anomalie-Detektoren auf EKG-Zeitreihen etabliert hat [48, 74]. Es handelt sich um einen EKG-Datensatz mit zwei Ableitungen, dessen Herzschläge manuell durch Experten gelabelt wurden. Der Datensatz der Charité verfügt lediglich über die Alarme des Monitoring-Systems, welche jedoch nicht zuverlässig sind. Daher wurde der MIT-BIH-Datensatz genutzt, um ein Modell mithilfe der Label zu entwickeln und auf den Charité-Datensatz zu evaluieren. Die Entwicklung des Modells orientierte sich an der Vorgehensweise von Ye et al. [74], da diese mit einem traditionellen Machine-Learning-Verfahren eine Accuracy von 99.77 % auf dem MIT-BIH-Datensatz erreichten, was im Vergleich zu anderen Studien ein sehr gutes Ergebnis darstellt. Da für dieses Projekt keine GPUs zu Verfügung standen, wurden keine Deep-Learning-Verfahren untersucht.

Die Implementierung der Erkennung von medizinischen Anomalien setzte sich aus mehreren Schritten zusammen: Die Elimination von technischen Artefakten, die Herzschlagerkennung und -segmentierung, die Feature-Extraktion und die Herzschlagklassifikation.

Aufgrund des entscheidenen Vorteils bezüglich des Rechenaufwands wurde die DWT als Entrauschungsprozedur zur automatisierten Bereinigung technischer Artefakte ausgewählt. Die Herschlagerkennung und -segmentierung wurde über ein festes Fenster durchgeführt, welches sich anhand des lokalisierten R-Peaks ausrichtet. Die Lokalisierung basierte auf einem Two-Moving-Average-Fenster nach Elgendi et al. [18]. Die Feature-Extraktion erfolgte über DWT-Koeffizienten und 14 identifizierte unabhängige Komponenten des Herzschlags. Auf Basis dieser Merkmale wurde eine Hauptkomponentenanalyse durchgeführt. Eine zehnfache Kreuzvalidierung zeigte bei 19 und 20 Haupkomponenten die größte Accuracy bei einer SVM als Klassifikator. Daher wurden 19 Hauptkomponenten für die weitere Modellent-

wicklung ausgewählt. Zusätzliche Feature bildeten vier verschiedene RR-Intervalle. Die Auswahl eines Klassifikators erfolgte zwischen einer LDA, einem KNN- und SVM-Klassifikator. Die SVM zeigte bei der Klassifikation von sechs verschiedenen Herzschlagtypen (fünf Arten von Anomalien und der normale Herzschlag) die beste Accuracy mit 99,13 %, Sensitivität von 99,75 % und den geringsten prozentualen Anteil von Falsch-Alarmen mit 0,62 %. Da bei den Daten der Charité bisher lediglich die Ableitungen nach Einthofen und Goldberger zu Verfügung standen, basierten die Modelle zur Weiterarbeit mit den Charité-Daten lediglich auf der ersten Ableitung im MIT-BIH-Datensatz. Diese entspricht in den meisten Fällen der zweiten Ableitung nach Einthofen. Auf Basis der zweiten Ableitung im MIT-BIH-Datensatz konnten die Ergebnisse auf eine Accuracy von 99,19 % gesteigert werden. Diese aufgezeichnete Ableitung entspricht jedoch meistens einer Ableitung nach Wilson, weshalb die Modelle auf Basis der zweiten Ableitung nicht weiter untersucht wurden. Die weitere Berechnung der Ableitungen V1 bis V6 nach Wilson auf den Charité-Daten könnte daher eine weitere Steigung bei der Anomalieerkennung geben.

Insgesamt konnte ein Prozess entwickelt werden, der mit einer Accuracy von 99,13 % erfolgreich und zuverlässig Anomalien auf dem MIT-BIH-Datensatz vorhersagt und damit relevante Stellen für den/die medizinischen Forscher*in auf dem EKG-Datensatz automatisiert feststellen kann. Die spezifischen DWT-Koeffizienten, die unabhängigen Komponenten des Herzschlags und die RR-Intervalle zeigen eine große Aussagekraft über das vorliegende EKG-Signal. Die Hauptkomponentenalyse ließ sich effektiv zur Dimensionsreduktion einsetzen.

Die daraufhin durchgeführte Übertragung auf den Charité-Datensatz zeigte in der ersten Implementierungs-Iteration mit einer Accuracy von 57,86 % noch keine zuverlässigen Ergebnisse. Die Datengrundlage zeigte zu unterschiedliche Eigenschaften. Die Übertragung eines Modell von externen Daten auf die Charité-Daten bedingt eine gleiche Auflösung oder einen zuverlässigeren Downsampling-Mechanismus. Aufgrund des Problems, dass die DWT sehr sensibel auf Zeitverschiebungen reagiert, wurde ein Verfahren ohne die DWT implementiert, was jedoch nur eine Accuracy von 57,86 % erbrachte. Durch dieselbe Vorverarbeitung inklusive einer Skalierung auf den selben Wertebereich, lag lediglich ein Unterschied in der Auflösung vor. Eine Weiterentwicklung des Vorgehens zur erfolgreichen automatisierten Erkennung von medizinischen Anomalien ist daher sehr zu empfehlen. Zusätzlich zu den in dieser Arbeit gewonnenen Erkenntnisse, können mögliche Verbesserungen aufgezeigt werden. Diese werden im folgenden Kapitel genauer erläutert.

6.2 AUSBLICK

Aufgrund der zu Ergebnisse des Modells auf den Charité-Daten, lassen sich einige mögliche Weiterentwicklungen ableiten. Zum einen könnte der implementierte Prozess verbessert werden, indem eine zuverlässige Möglich-

keit des Downsamplings der Daten geschaffen wird. Zum anderen könnte mithilfe anderer Algorithmen, wie beispielsweise welche mit einem Semi-Supervised-Ansatz, das Finden von Anomalien im Datensatz verbessert werden. Bei einem Semi-Supervised-Ansatz wird anhand der normalen Herzschläge gelernt und daraufhin jede Abweichung von der Norm als Anomalie gesehen. Dies könnte zu einer zuverlässigeren Erkennung von verschiedenen Anomalien führen und würde keinen vollständig gelabelten Datensatz mit allen vorkommenden Anomalien voraussetzen, sondern lediglich Daten, die der normalen Struktur entsprechen, nutzen. Inbesondere ist hier die Bedeutung der Bereinigung von den EKG-Daten weiterhin herauszustellen, da unbereinigte Daten ebenfalls eine andere Struktur als ein normaler Herzschlag aufweisen und daher ebenfalls als Anomalie gesehen würden. Zudem würde die Forschung durch das medizinische Personal aufgrund der schlechten Lesbarkeit erschwert werden.

Die Hinzunahme der Ableitungen nach Wilson könnte ebenfalls eine vielversprechende Möglichkeit der Weiterentwicklung sein, da sie die Ausbreitung des elektrischen Signals aus einem anderen Winkel betrachten und insbesondere für Anomalien eine stärker abweichende Morphologie von der Norm erreichen könnten. Auch die Forschung der mulivariaten Betrachtung, indem beispielsweise der Verlauf der Blutdruckkurve zusätzlich miteinbezogen wird, könnte entscheidende Informationen zur Klassifikation erbringen. Sun et al. [67] zeigten bereits, dass die Kombination der arteriellen Blutdruck-Kurve und des EKG-Signals ein vielversprechender Lösungsansatz sein könnte.

LITERATUR

[1] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen und Shen Luo. "ECG beat detection using filter banks". In: *IEEE Transactions on Biomedical Engineering* 46.2 (1999), S. 192–202. DOI: 10.1109/10.740882.

[2] Charu C. Aggarwal. *Outlier Analysis*. 2nd. Springer Publishing Company, Incorporated, 2017. ISBN: 3319475770.

[3] Malik Agyemang, Ken Barker und Rada Alhajj. "A Comprehensive Survey of Numeric and Symbolic Outlier Mining Techniques". In: *Intell. Data Anal.* 10.6 (Dez. 2006). ISSN: 1088-467X.

[4] Silvia Maria Alessio. "Discrete Wavelet Transform (DWT)". In: *Digital Signal Processing and Spectral Analysis for Scientists: Concepts and Applications*. Cham: Springer International Publishing, 2016, S. 645–714. ISBN: 978-3-319-25468-5. DOI: 10.1007/978-3-319-25468-5_14. URL: https://doi.org/10.1007/978-3-319-25468-5_14.

[5] Maheswari Arumugam und Arun Kumar Sangaiah. "Arrhythmia identification and classification using wavelet centered methodology in ECG signals". In: *Concurrency and Computation: Practice and Experience* 32 (Nov. 2019). DOI: 10.1002/cpe.5553.

[6] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf und Gunnar Rätsch. "Support Vector Machines and Kernels for Computational Biology". In: *PLoS computational biology* 4 (Nov. 2008), e1000173. DOI: 10.1371/journal.pcbi.1000173.

[7] Manfred Borchert. *ELEKTROKARDIOGRAMM Handbuch für Einsteiger*. 2005. URL: https://www.nhhberlin.de/assets/files/EKG_Buch_01.pdf.

[8] J. Nathan Brunton Steven L. und Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. 1st. USA: Cambridge University Press, 2019. ISBN: 1108422098.

[9] Statistisches Bundesamt. *Todesursachen*. https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Todesursachen/_inhalt.html. Accessed: 2021-01-26.

[10] George E. Burch. "The history of vectorcardiography". In: *Medical History* 29.S5 (1985). DOI: 10.1017/S002572730007054X.

[11] E. J. Candes, J. Romberg und T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on Information Theory* 52.2 (2006), S. 489–509. DOI: 10.1109/TIT.2005.862083.

[12] Varun Chandola, Arindam Banerjee und Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3 (Juli 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: https://doi.org/10.1145/1541880.1541882.

[13] Szi-Wen Chen, Hsiao-Chen Chen und Hsiao-Lung Chan. "A real-time QRS detection method based on moving-averaging incorporating with wavelet denoising". In: *Computer Methods and Programs in Biomedicine* 82.3 (2006), S. 187 –195. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2005.11.012.

[14] Samjin Choi, Mourad Adnane, Gi-Ja Lee, Hoyoung Jang, Zhongwei Jiang und Hun-Kuk Park. "Development of ECG beat segmentation method by combining lowpass filter and irregular R–R interval checkup strategy". In: *Expert Systems with Applications* 37.7 (2010), S. 5208 –5218. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2009.12.069.

[15] Ivaylo Christova, Gèrman Gómez-Herrero, Vessela Krasteva, Irena Jekova, Atanas Gotchev und Karen Egiazarian. "Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification". In: *Medical Engineering Physics* 28.9 (2006). ISSN: 1350-4533. DOI: 10.1016/j.medengphy.2005.12.010.

[16] David Donoho und Michael Elad. "Optimally sparse representation from overcomplete dictionaries via l1 norm minimization". In: *Proc. Natl. Acad. Sci. USA* 100 (Jan. 2003), S. 2002–2197. DOI: https://doi.org/10.1073/pnas.0437847100.

[17] Bogdan Dumitrescu und Paul Irofti. *Dictionary Learning Algorithms and Applications*. 1st. Springer Publishing Company, Incorporated, 2018. ISBN: 3319786733. DOI: https://doi.org/10.1007/978-3-319-78674-2.

[18] Mohamed Elgendi, Mirjam Jonkman und Friso DeBoer. "Frequency Bands Effects on QRS Detection." In: Jan. 2010, S. 428–431. DOI: 10.5220/0002742704280431.

[19] Apache Software Foundation. *Spark Python API Docs 3.0.0*. URL: https://spark.apache.org/docs/latest/sql-pyspark-pandas-with-arrow.html# (besucht am 20. 08. 2020).

[20] R. Muthukrishnan und G. Poonkuzhali. "A Comprehensive Survey on Outlier Detection Methods". In: *American-Eurasian Journal of Scientific Research* 12.3 (2017). URL: https://idosi.org/aejsr/12(3)17/7.pdf.

[21] Said Gaci. "A New Ensemble Empirical Mode Decomposition (EEMD) Denoising Method for Seismic Signals". In: *Energy Procedia* 97 (Nov. 2016), S. 84–91. DOI: 10.1016/j.egypro.2016.10.026.

[22] Victor Giurgiutiu. "Chapter 14 - Signal Processing and Pattern Recognition for Structural Health Monitoring with PWAS Transducers". In: *Structural Health Monitoring with Piezoelectric Wafer Active Sensors (Second Edition)*. Hrsg. von Victor Giurgiutiu. Second Edition. Oxford: Academic Press, 2014, S. 807 –862. ISBN: 978-0-12-418691-0. DOI: https:

//doi.org/10.1016/B978-0-12-418691-0.00014-9. URL: http://www.sciencedirect.com/science/article/pii/B9780124186910000149.

[23] Will Gragido, Johnl Pirc, Nick Selby und Daniel Molina. "Chapter 4 - Signal-to-Noise Ratio". In: *Blackhatonomics*. Hrsg. von N. Selby W. Gragido J. Pirc und D. Molina. Boston: Syngress, 2013, S. 45 –55. ISBN: 978-1-59749-740-4. DOI: https://doi.org/10.1016/B978-1-59-749740-4.00004-6. URL: http://www.sciencedirect.com/science/article/pii/B9781597497404000046.

[24] Manish Gupta, Jing Gao, Charu Aggarwal und Jiawei Han. *Outlier Detection for Temporal Data*. Morgan Claypool Publishers, 2014. ISBN: 1627053751.

[25] Isabelle Guyon, Masoud Nikravesh, Steve Gunn und Lotfi A. Zadeh. *Feature Extraction Foundations and Applications*. Springer, Berlin, Heidelberg, 2006. ISBN: 978-3-540-35488-8. DOI: https://doi.org/10.1007/978-3-540-35488-8.

[26] A. Kurtz und S. Silbernagl H. Pape. *Physiologie*. 7. Georg Thieme Verlag KG, 2014. ISBN: ISBN 978-3-13-796007-2.

[27] Trevor Hastie, Robert Tibshirani und Jerome Friedman. "Linear Methods for Regression". In: *The Elements of Statistical Learning*. New York: Springer-Verlag New York, 2009. ISBN: 978-0-387-84858-7. DOI: https://doi.org/10.1007/978-0-387-84858-7.

[28] D. M. Hawkins. *Identification of outliers*. London: Chapman und Hall, 1980. ISBN: 978-94-015-3994-4. DOI: https://doi.org/10.1007/978-94-015-3994-4.

[29] Meng He, Liu Feng und Dongdong Zhao. "A method to enhance SNR based on CEEMDAN and the interval thresholding in $_OTDRsystems$". In: *Applied Physics B* 126 (Mai 2020). DOI: 10.1007/s00340-020-07448-x.

[30] Victoria Hodge und Jim Austin. "A Survey of Outlier Detection Methodologies". In: *Artif. Intell. Rev.* 22.2 (Okt. 2004). ISSN: 0269-2821. DOI: 10.1023/B:AIRE.0000045502.10941.a9. URL: https://doi.org/10.1023/B:AIRE.0000045502.10941.a9.

[31] Steven M. Holland. "Principal Components Analysis (PCA)". In: (2019). ISSN: 0269-2821. URL: http://strata.uga.edu/software/pdf/pcaTutorial.pdf.

[32] Aapo Hyvärinen. "Survey on Independent Component Analysis". In: *Neural Computing Surveys* 2 (1999), S. 94–128. URL: https://www.cs.helsinki.fi/u/ahyvarin/papers/NCS99.pdf.

[33] Gareth James, Daniela Witten, Trevor Hastie und Robert Tibshirani. "Principal Components Analysis". In: *An Introduction to Statistical Learning*. New York: Springer Science+Business Media New York, 2013, S. 374–385. ISBN: 978-1-4614-7138-7. DOI: https://doi.org/10.1007/978-1-4614-7138-7.

[34] Pramendra Kumar und Vijay Kumar Sharma. "Detection and classification of ECG noises using decomposition on mixed codebook for quality analysis". In: *Healthcare Technology Letters* 7.1 (2020), S. 18–24. DOI: 10.1049/htl.2019.0096.

[35] Zhu Kunpeng, Wong Yoke San und Hong Geok Soon. "4 - Signal processing for tool condition monitoring: from wavelet analysis to sparse decomposition". In: *Mechatronics and Manufacturing Engineering*. Hrsg. von J. Paulo Davim. Woodhead Publishing Reviews: Mechanical Engineering Series. Woodhead Publishing, 2012, S. 115 –157. ISBN: 978-0-85709-150-5. DOI: https://doi.org/10.1533/9780857095893.115. URL: http://www.sciencedirect.com/science/article/pii/B9780857091505500042.

[36] K. N. Leach. "A survey paper on independent component analysis". In: *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No.02EX540)*. 2002, S. 239–242. DOI: 10.1109/SSST.2002.1027042.

[37] Hong Zu und Pierre Boulanger Li. "A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG)". In: *Sensors* 20.5 (2020). DOI: 10.3390/s20051461.

[38] Kang Li, Nan Du und Aidong Zhang. "Detecting ECG Abnormalities via Transductive Transfer Learning". In: *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. BCB '12. Orlando, Florida: Association for Computing Machinery, 2012. ISBN: 9781450316705. DOI: 10.1145/2382936.2382963. URL: https://doi.org/10.1145/2382936.2382963.

[39] Mariano Llamedo und Juan Pablo Martínez. "Heartbeat Classification Using Feature Selection Driven by Database Generalization Criteria". In: *IEEE Transactions on Biomedical Engineering* 58.3 (2011), S. 616–625. DOI: 10.1109/TBME.2010.2068048.

[40] *MIT-BIH Arrhythmias Database*. 2005. URL: https://physionet.org/content/mitdb/1.0.0/ (besucht am 06. 01. 2021).

[41] M. Sabarimalai Manikandana und K. P. Soman. "A novel method for detecting R-peaks in electrocardiogram (ECG) signal". In: *Biomedical Signal Processing and Control* 7.2 (2012), S. 118 –128. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2011.03.004.

[42] Wilhelm Burger und Mark James Burge. "Die diskrete Kosinustransformation (DCT)". In: *Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, S. 367–374. ISBN: 978-3-540-27653-1. DOI: 10.1007/3-540-27653-X_15. URL: https://doi.org/10.1007/3-540-27653-X_15.

[43] Wilhelm Burger und Mark James Burge. "Einführung in Spektraltechniken". In: *Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, S. 479–509. ISBN: 978-3-642-04604-9. DOI: 10.1007/978-3-642-04604-9_18. URL: https://doi.org/10.1007/978-3-642-04604-9_18.

[44] Matplotlib. *matplotlib.pyplot.psd*. https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.psd.html. Accessed: 2021-01-27.

[45] Alfred Mertins. "Filterbänke". In: *Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. Wiesbaden: Vieweg+Teubner, 2010, S. 169–226. ISBN: 978-3-8348-9381-9. DOI: 10.1007/978-3-8348-9381-9_7. URL: https://doi.org/10.1007/978-3-8348-9381-9_7.

[46] Alfred Mertins. "Die Wavelet-Transformation". In: *Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2013, S. 301–354. ISBN: 978-3-8348-8109-0. DOI: 10.1007/978-3-8348-8109-0_9. URL: https://doi.org/10.1007/978-3-8348-8109-0_9.

[47] Fatiha Bouaziz; Daoud Boutana und Messaoud Benidir. "Multiresolution wavelet-based QRS complex detection algorithm suited to several abnormal morphologies". In: *IET Signal Processing* 8.7 (2014). DOI: https://doi.org/10.1049/iet-spr.2013.0391.

[48] George B. Moody und Roger G. Mark. "The impact of the MIT-BIH Arrhythmia Database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), S. 45–50. DOI: 10.1109/51.932724.

[49] Meinard Müller. "Dynamic Time Warping". In: *Information Retrieval for Music and Motion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 69–84. ISBN: 978-3-540-74048-3. DOI: 10.1007/978-3-540-74048-3_4. URL: https://doi.org/10.1007/978-3-540-74048-3_4.

[50] Thanh-Nghia Nguyen. "Artifact elimination in ECG signal using wavelet transform". In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 18 (März 2020), S. 936–944. DOI: 10.12928/telkomnika.v18i2.14403.

[51] Jiapu Pan und Willis J. Tompkins. "A Real-Time QRS Detection Algorithm". In: *IEEE Transactions on Biomedical Engineering* BME-32.3 (1985), S. 230–236. DOI: 10.1109/TBME.1985.325532.

[52] Physionet.org. *PhysioBank Annotations*. 2016. URL: https://archive.physionet.org/physiobank/annotations.shtml (besucht am 06. 01. 2021).

[53] PyEMD. *PyEMD's documentation*. https://pyemd.readthedocs.io/en/latest/. Accessed: 2021-01-27.

[54] PyWavelets. *PyWavelets - Wavelet Transforms in Python*. https://pywavelets.readthedocs.io/en/latest/. Accessed: 2021-01-27.

[55] Holden Karau und Rachel Warren. *High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark*. 1st. O'Reilly Media, Inc., 2017. ISBN: 1491943203.

[56] Meisam Razaviyayn, Hung-Wei Tseng und Zhi-Quan Luo. "Dictionary learning for sparse representation: Complexity and algorithms". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, S. 5247–5251. DOI: 10.1109/ICASSP.2014.6854604.

[57] Philip de Chazal und Richard B. Reilly. "A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features". In: *IEEE Transactions on Biomedical Engineering* 53.12 (2006). ISSN: 0018-9294. DOI: 10.1109/tbme.2006.883802.

[58] Robert-Koch-Institut. *Sterblichkeit und Todesursachen*. https://www.rki.de/DE/Content/Gesundheitsmonitoring/Themen/Demografischer_Wandel/Sterblichkeit/Sterblichkeit_node.html. Accessed: 2021-01-26.

[59] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller und Marius Kloft. "Deep Semi-Supervised Anomaly Detection". In: *CoRR* abs/1906.02694 (2019). arXiv: 1906.02694. URL: http://arxiv.org/abs/1906.02694.

[60] Udit Satija, Madhusmita Mohanty und Barathram Ramkumar. "Cyclostationary Features Based Modulation Classification in Presence of Non Gaussian Noise Using Sparse Signal Decomposition". In: *Wirel. Pers. Commun.* 96.4 (2017). DOI: 10.1007/s11277-017-4444-4.

[61] Udit Satija, Barathram Ramkumar und M. Sabarimalai Manikandan. "A unified sparse signal decomposition and reconstruction framework for elimination of muscle artifacts from ECG signal". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, S. 779–783. DOI: 10.1109/ICASSP.2016.7471781.

[62] Udit Satija, Barathram Ramkumar und M. Sabarimalai Manikandan. "Noise-aware dictionary-learning-based sparse representation framework for detection and removal of single and combined noises from ECG signal". In: *Healthcare Technology Letters* 4.1 (2017), S. 2–12. DOI: 10.1049/htl.2016.0077.

[63] Shubhankar Saxena, Rohan Jais und Malaya Kumar Hota. "Removal of Powerline Interference from ECG Signal using FIR, IIR, DWT and NLMS Adaptive Filter". In: *2019 International Conference on Communication and Signal Processing (ICCSP)*. 2019, S. 0012–0016. DOI: 10.1109/ICCSP.2019.8698112.

[64] K. K. Shukla und K. Arvind Tiwari. "Chapter 2 - Filter Banks and DWT". In: *Efficient Algorithms for Discrete Wavelet Transform - With Applications to Denoising and Fuzzy Inference Systems*. Springer, 2013. ISBN: 978-1-4471-4940-8. DOI: https://doi.org/10.1007/978-1-4471-4941-5.

[65] Karl Siebertz, David van Bebber und Thomas Hochkirchen. "Komponentenanalyse". In: *Statistische Versuchsplanung: Design of Experiments (DoE)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, S. 395–414.

ISBN: 978-3-662-55743-3. DOI: 10.1007/978-3-662-55743-3_12. URL: https://doi.org/10.1007/978-3-662-55743-3_12.

[66] Brij N. Singh und Arvind K. Tiwari. "Optimal selection of wavelet basis function applied to ECG signal denoising". In: *Digital Signal Processing* 16.3 (2006), S. 275 –287. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2005.12.003. URL: http://www.sciencedirect.com/science/article/pii/S1051200405001703.

[67] J. X. Sun, A. T. Reisner und R. G. Mark. "A signal abnormality index for arterial blood pressure waveforms". In: (2006), S. 13–16. URL: https://ieeexplore.ieee.org/document/4511776.

[68] Manu Thomas, Manab Kr Das und Samit Ari. "Automatic ECG arrhythmia classification using dual tree complex wavelet based features". In: *AEU - International Journal of Electronics and Communications* 69.4 (2015), S. 715 –721. ISSN: 1434-8411. DOI: https://doi.org/10.1016/j.aeue.2014.12.013.

[69] María E. Torres, Marcelo A. Colominas, Gastón Schlotthauer und Patrick Flandrin. "Complete ensemble empirical mode decomposition with adaptive noise". In: Mai 2011, S. 4144–4147. DOI: 10.1109/ICASSP.2011.5947265.

[70] Bharadwaj Veeravalli, Chacko John Deepu und DuyHoa Ngo. "Real-Time, Personalized Anomaly Detection in Streaming Data for Wearable Healthcare Devices". In: *Handbook of Large-Scale Distributed Computing in Smart Healthcare*. Hrsg. von Samee U. Khan, Albert Y. Zomaya und Assad Abbas. Cham: Springer International Publishing, 2017, S. 403–426. ISBN: 978-3-319-58280-1. DOI: 10.1007/978-3-319-58280-1_15. URL: https://doi.org/10.1007/978-3-319-58280-1_15.

[71] Aswathy Velayudhan1 und Soniya Peter. "Noise Analysis and Different Denoising Techniques of ECG Signal - A Survey". In: 2016. URL: http://iosrjournals.org/iosr-jece/papers/ICETEM/Vol.\%201\%20Issue\%201/ECE\%2006-40-44.pdf.

[72] Zhaohua Wu und Norden E. Huang. "Ensemble Empirical Mode Decomposition: a Noise-Assisted Data Analysis Method". In: *Adv. Data Sci. Adapt. Anal.* 1 (2009), S. 1–41. DOI: https://doi.org/10.1142/S1793536909000047.

[73] Yang Xu, Mingzhang Luo, Tao Li und Gangbing Song. "ECG Signal De-noising and Baseline Wander Correction Based on CEEMDAN and Wavelet Threshold". In: *Sensors (Basel, Switzerland)* 17 (2017). DOI: 10.3390/s17122754.

[74] Can Ye, B. V. K. Vijaya Kumar und Miguel Tavares Coimbra. "Heartbeat Classification Using Morphological and Dynamic Features of ECG Signals". In: *IEEE Transactions on Biomedical Engineering* 59.10 (2012), S. 2930–2941. DOI: 10.1109/TBME.2012.2213253.

[75] A. Zeiler, R. Faltermeier, I. R. Keck, A. M. Tomé, C. G. Puntonet und E. W. Lang. "Empirical Mode Decomposition - an introduction". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. 2010, S. 1–8. DOI: 10.1109/IJCNN.2010.5596829.

[76] Zahia Zidelmal, Ahmed Amirou, Mourad Adnane und Adel Belouchrani. "QRS detection based on wavelet coefficients". In: *Computer Methods and Programs in Biomedicine* 107.3 (2012), S. 490 –496. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2011.12.004.

[77] J. C. Behrends et al. *Duale Reihe Physiologie*. 3. Georg Thieme Verlag KG, 2017. ISBN: 978-3-13-138413-3.

Teil II

APPENDIX

A

ELIMINATION TECHNISCHER ARTEFAKTE

Zunächst werden alle Artefakte, welche mithilfe von Sinus- und Kosinuswellen simuliert wurden, dargestellt. Pro Artefakt (Baseline Wander (BW), Power-Line-Interferenz (PLI) und BW+PLI) wurden drei Ausführungen erstellt, die einem unterschiedlichen Schweregrad entsprechen. Jedes dieser Artefakte wurde additiv zu den einzelnen rauschfreien Signalen hinzugefügt, um so ein EKG-Signal inklusive Artefakt zu simulieren. Abbildung A.1 zeigt die drei Artefakte des BWs. In Abbildung A.2 werden die drei Artefakte des PLIs dargestellt und Abbildung A.3 entspricht der Visualisierung der Kombination aus BW und PLI in drei Ausführungen.

A.1 SIMULATIONEN DER TECHNISCHEN ARTEFAKTE

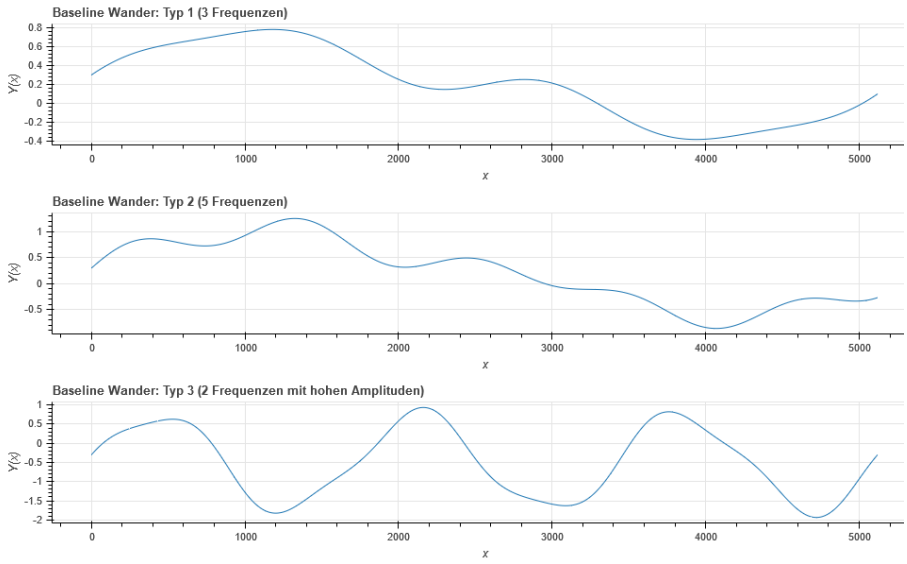

Abbildung A.1: Simuliertes Baseline Wander zur Evaluation der Entrauschungsmethoden

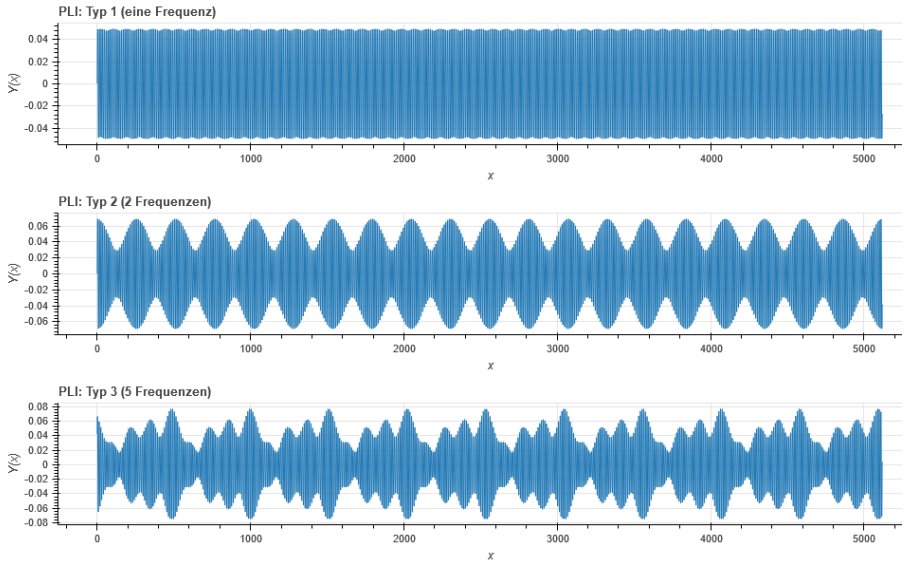

Abbildung A.2: Simulierte Power-Line-Interferenz zur Evaluation der Entrauschungsmethoden

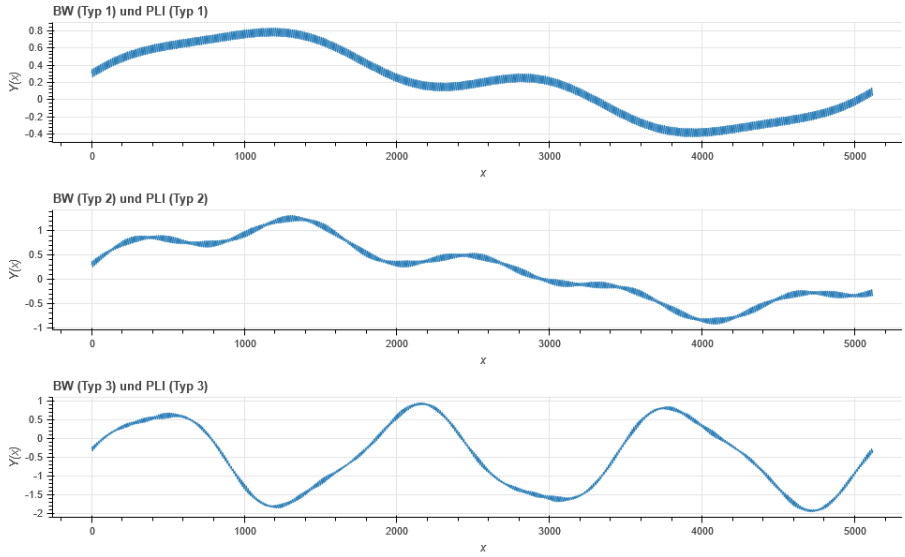

Abbildung A.3: Kombination von Baseline Wander und Power-Line-Interferenz zur Evaluation der Entrauschungsmethoden

A.2 QUELLCODE DES METHODENVERGLEICHS

Der nachfolgende Code-Ausschnitt zeigt die einzelnen gegenübergestellten Entrauschungsmethoden.

```
def mean_squared_error(y_clean, y_reconst):
    return 1/len(y_clean) * np.sum(np.square(y_clean-y_reconst))

def normalised_cross_correlation(y_clean, y_reconst):
```

```
    return np.correlate(y_clean-np.mean(y_clean), y_reconst-np.mean(y_reconst))
    /(np.sqrt(np.sum(np.square(y_clean-np.mean(y_clean)))*np.sum(np.square(
    y_reconst-np.mean(y_reconst)))))

def maximum_absolute_error(y_clean, y_reconst):
    return np.max(np.abs(y_clean - y_reconst))

def signal_noise_ratio(y_clean, y_reconst):
    return 10 * np.log10(np.sum(np.square(y_clean-np.mean(y_clean)))/np.sum(np.
    square(y_clean-y_reconst)))

def sparse_signal_decomposition(y_noise):

    #Create Sinus- & Cosinus-Transformation Matrix
    s_ij = np.fromfunction(lambda i, j: np.sqrt(2/P)*(1* np.sin((np.pi*(2*j + 1)
    *(i+1))/(2*P))), (P-1,P))
    last_row_s = np.array([np.sqrt(2/P)*(np.sqrt(1/2)* np.sin((np.pi*(2*j + 1)*(
    P+1))/(2*P))) for j in range(P)])
    s_ij = np.vstack([s_ij, last_row_s])
    c_ij = np.fromfunction(lambda i, j: np.sqrt(2/P)*(1* np.cos((np.pi*(2*j + 1)
    *(i))/(2*P))), (P,P))
    first_row_c = np.array([np.sqrt(2/P)*(np.sqrt(1/2)* np.cos((np.pi*(2*j + 1)
    *(0))/(2*P))) for j in range(P)])
    c_ij = np.vstack([first_row_c, c_ij[1:]])

    #Extraction of Baseline Wander & PLI frequencies
    bw_s = s_ij[:,0:20]
    bw_c = c_ij[:,0:20]
    pli_s = s_ij[:,900:1060]
    pli_c = c_ij[:,900:1060]

    #Create Phi Matrix (Combining of sine- and cosine transformation vectors)
    phi_bw = np.append(bw_c, bw_s, axis=1)
    phi_pli = np.append(pli_c, pli_s, axis=1)
    phi = np.append(phi_bw, phi_pli, axis=1)

    #Convex l1 Optimization
    p = cp.Variable(360)
    objective = cp.Minimize((cp.sum_squares((phi@p) - y_noise))+ 0.1*cp.norm(p,
    1))
    prob = cp.Problem(objective)

    result = prob.solve()
    p_esteem = p.value

    #Estimation of PLI and BW
    y_bw = phi_bw.dot(p_esteem[0:40])
    y_pli = phi_pli.dot(p_esteem[40:360])

    #Subtract noise of original ECG Signal
    return y_noise - y_bw - y_pli

def discrete_wavelet_decomposition(wavelet_type, y_noise):
    DWTcoeffs = pywt.wavedec(y_noise,wavelet_type,mode='symmetric', level=8,
    axis=-1)
    DWTcoeffs[0] = np.zeros_like(DWTcoeffs[0])
    DWTcoeffs[-1] = np.zeros_like(DWTcoeffs[-1])
```

```
    DWTcoeffs[-2] = np.zeros_like(DWTcoeffs[-2])
    DWTcoeffs[-3] = np.zeros_like(DWTcoeffs[-3])
    return pywt.waverec(DWTcoeffs,wavelet_type,mode='symmetric',axis=-1)

def ceemdan_with_dwt(y_noise):
    ceemdan = CEEMDAN()
    cIMFs = ceemdan(y_noise)

    for i in np.arange(0,(len(cIMFs)-1)):
        DWTcoeffs = pywt.wavedec(cIMFs[i],"dmey",mode='symmetric', level=8, axis
    =-1)
        DWTcoeffs[0] = np.zeros_like(DWTcoeffs[0])
        DWTcoeffs[-1] = np.zeros_like(DWTcoeffs[-1])
        DWTcoeffs[-2] = np.zeros_like(DWTcoeffs[-2])
        DWTcoeffs[-3] = np.zeros_like(DWTcoeffs[-3])
        cIMFs[i] = pywt.waverec(DWTcoeffs,"dmey",mode='symmetric',axis=-1)

    return np.sum(cIMFs[2:], axis=0)
```

Listing A.1: Methoden zur Elimination der technischen Artefakte

Nachdem die Methoden zur Bereinigung definiert wurden, können mithilfe einer Evaluationsmethode die in der Arbeit genutzten Metriken SNR, MAX, MSE und NCC auf die Methoden anschließend angewendet werden. Der nachfolgende Code-Ausschnitt A.2 entspricht der Methode zur Evaluation der Zeitreihen-Bereinigung.

```
#Evaluation method
def evaluation(y_noise, range_begin, range_end, bw, pli):
    evaluation_schema = [
        StructField("c_testset_id", IntegerType(), False),
        StructField("c_channel", StringType(), False),
        StructField("c_method", StringType(), False),
        StructField("c_bw_type", StringType(), True),
        StructField("c_pli_type", StringType(), True),
        StructField("c_snr", DoubleType(), False),
        StructField("c_max", DoubleType(), False),
        StructField("c_ncc", DoubleType(), False),
        StructField("c_mse", DoubleType(), False),
    ]
    schema = StructType(evaluation_schema)
    evaluation_df = sqlContext.createDataFrame(sc.emptyRDD(), schema)

    for i in range(range_begin, range_end):
        #Get clean ecg signal
        y = np.asarray(testset.select("c_data_array").where(col("c_id") == i).
    collect()).flatten()
        y_clean = y-np.mean(y)
        y_with_noise = y_noise + y_clean

        #Sparse signal decomposition
        y_clean_ssd_ = sparse_signal_decomposition(y_with_noise)
        y_clean_ssd = y_clean_ssd_-np.mean(y_clean_ssd_)

        #Discrete Wavelet Decomposition
        y_clean_dwt_db4_ = discrete_wavelet_decomposition('db4', y_with_noise)
        y_clean_dwt_db4 = y_clean_dwt_db4_-np.mean(y_clean_dwt_db4_)
```

```
        y_clean_dwt_dmey_ = discrete_wavelet_decomposition('dmey', y_with_noise)
        y_clean_dwt_dmey = y_clean_dwt_dmey_-np.mean(y_clean_dwt_dmey_)

        #CEEMDAN with DWT
        y_clean_ceemdan_ = ceemdan_with_dwt(y_with_noise)
        y_clean_ceemdan = y_clean_ceemdan_-np.mean(y_clean_ceemdan_)

        #Calculation evaluation metrics
        snr_ssd = float(signal_noise_ratio(y_clean, y_clean_ssd))
        snr_dwt_db4 = float(signal_noise_ratio(y_clean, y_clean_dwt_db4))
        snr_dwt_dmey = float(signal_noise_ratio(y_clean, y_clean_dwt_dmey))
        snr_ceemdan = float(signal_noise_ratio(y_clean, y_clean_ceemdan))

        max_ssd = float(maximum_absolute_error(y_clean, y_clean_ssd))
        max_dwt_db4 = float(maximum_absolute_error(y_clean, y_clean_dwt_db4))
        max_dwt_dmey = float(maximum_absolute_error(y_clean, y_clean_dwt_dmey))
        max_ceemdan = float(maximum_absolute_error(y_clean, y_clean_ceemdan))

        ncc_ssd = float(normalised_cross_correlation(y_clean, y_clean_ssd))
        ncc_dwt_db4 = float(normalised_cross_correlation(y_clean,
    y_clean_dwt_db4))
        ncc_dwt_dmey = float(normalised_cross_correlation(y_clean,
    y_clean_dwt_dmey))
        ncc_ceemdan = float(normalised_cross_correlation(y_clean,
    y_clean_ceemdan))

        mse_ssd = float(mean_squared_error(y_clean, y_clean_ssd))
        mse_dwt_db4 = float(mean_squared_error(y_clean, y_clean_dwt_db4))
        mse_dwt_dmey = float(mean_squared_error(y_clean, y_clean_dwt_dmey))
        mse_ceemdan = float(mean_squared_error(y_clean, y_clean_ceemdan))

        #get param
        channel = testset.select("c_channel").where(col("c_id") == i).collect()
    [0]["c_channel"]

        newRows = spark.createDataFrame([(i, channel, "SSD", bw, pli, snr_ssd,
    max_ssd, ncc_ssd, mse_ssd), (i, channel, "DWT-db4", bw, pli, snr_dwt_db4,
    max_dwt_db4, ncc_dwt_db4, mse_dwt_db4), (i, channel, "DWT-dmey", bw, pli,
    snr_dwt_dmey, max_dwt_dmey, ncc_dwt_dmey, mse_dwt_dmey), (i, channel, "
    CEEMDAN", bw, pli, snr_ceemdan, max_ceemdan, ncc_ceemdan, mse_ceemdan)],
    schema)
        evaluation_df = evaluation_df.union(newRows)

    return evaluation_df
```

Listing A.2: Methode zur Evaluation der einzelnen Entrauschungsmethoden

Zum Schluss werden die in Abbildung A.1, A.2 und A.3 definierten Artefakte erstellt, zu den EKG-Signalen hinzugefügt und die Evaluationsmethode aufgerufen.

```
#Different technical artefacts
P = 5120
Fs = 512
x = np.arange(P)
Fs = 512
f1 = 0.1
```

```
f2 = 0.2
f3 = 0.35
y_bw_1 = (np.sin(2 * np.pi * f1 * x / Fs)*0.5 + 0.2) + (np.sin(2 * np.pi * f2 *
    x / Fs)*0.2) + (np.cos(2 * np.pi * f3 * x / Fs)*0.1)

evaluation_df = evaluation(y_bw_1, 1, 11, "BW Typ1", "")
f1 = 0.1
f2 = 0.2
f3 = 0.35
f4 = 0.47
f5 = 0.09
y_bw_2 = (np.sin(2 * np.pi * f1 * x / Fs)*0.5 + 0.2) + (np.sin(2 * np.pi * f2 *
    x / Fs)*0.2) + (np.cos(2 * np.pi * f3 * x / Fs)*0.1) + (np.sin(2 * np.pi *
    f4 * x / Fs)*0.2) + (np.sin(2 * np.pi * f5 * x / Fs)*0.3)
evaluation_df = evaluation_df.union(evaluation(y_bw_2, 11, 21, "BW Typ2", ""))

f1 = 0.3
f2 = 0.7
y_bw_3 = (np.sin(2 * np.pi * f1 * x / Fs)*1.25 - 0.5) + (np.cos(2 * np.pi * f2 *
     x / Fs)*0.2 )
evaluation_df = evaluation_df.union(evaluation(y_bw_3, 21, 31, "BW Typ3", ""))

f1 = 46
y_pli_1 = np.sin(2 * np.pi * f1 * x / Fs) * 0.05
evaluation_df = evaluation_df.union(evaluation(y_pli_1, 1, 11, "", "PLI Typ1"))

f1 = 48
f2 = 46
y_pli_2 = (np.sin(2 * np.pi * f1 * x / Fs) * 0.02) + (np.sin(2 * np.pi * f2 * x
    / Fs) * 0.05)
evaluation_df = evaluation_df.union(evaluation(y_pli_2, 11, 21, "", "PLI Typ2"))

f1 = 48
f2 = 46
f3 = 47
f4 = 50
f5 = 47
y_pli_3 = ((np.sin(2 * np.pi * f1 * x / Fs) * 0.02) + (np.sin(2 * np.pi * f2 * x
     / Fs) * 0.05) + (np.sin(2 * np.pi * f3 * x / Fs) * 0.01) + (np.cos(2 * np.
    pi * f4 * x / Fs) * 0.02) + (np.cos(2 * np.pi * f5 * x / Fs) * 0.04))*0.7
evaluation_df = evaluation_df.union(evaluation(y_pli_3, 21, 31, "", "PLI Typ3"))

evaluation_df = evaluation_df.union(evaluation(y_bw_1+y_pli_1, 1, 11, "BW Typ1",
     "PLI Typ1"))
evaluation_df = evaluation_df.union(evaluation(y_bw_2+y_pli_2, 11, 21, "BW Typ2"
    , "PLI Typ2"))
evaluation_df = evaluation_df.union(evaluation(y_bw_3+y_pli_3, 21, 31, "BW Typ3"
    , "PLI Typ3"))
```

Listing A.3: Aufruf der Evaluationsmethode nach Erstellung der Signale mit entsprechendem Artefakt

A.3 ERGEBNISSE DER EVALUATION

Die Ergebnisse der Evaluationsmethode werden in den folgenden zwei Tabellen dargestellt. Zu jedem Artefakt wurden alle vier Bereinigungsmethoden über die Metriken evaluiert. Die Tabellen A.1 und A.2 zeigen das arithmetische Mittel aller evaluierten EKG-Zeitreihen.

Artefakt	**SSD**				**CEEMDAN**			
	SNR	**MAX**	**NCC**	**MSE**	**SNR**	**MAX**	**NCC**	**MSE**
BW	12,483	0,151	0,957	0,003	-4,492	0,443	0,465	0,156
PLI	12,920	0,130	0,960	0,003	7,579	0,442	0,898	0,898
BW+PLI	12,428	0,153	0,957	0,003	-3,28	0,446	0,479	0,479

Tabelle A.1: Ergebnisse der SSD und CEEMDAN für den SNR, MAX, NCC und MSE bezüglich der Fähigkeit zur Elimination technischer Artefakte

Artefakt	**DWT-db4**				**DWT-dmey**			
	SNR	**MAX**	**NCC**	**MSE**	**SNR**	**MAX**	**NCC**	**MSE**
BW	8,669	0,443	0,915	0,006	10,342	0,308	0,938	0,004
PLI	9,347	0,442	0,930	0,005	10,597	0,302	0,942	0,004
BW+PLI	8,586	0,446	0,914	0,006	10,342	0,308	0,938	0,004

Tabelle A.2: Ergebnisse der DWT mit dem dmey- und db4-Wavelet für den SNR, MAX, NCC und MSE bezüglich der Fähigkeit zur Elimination technischer Artefakte

B

FINDEN VON MEDIZINISCHEN ANOMALIEN

Die Entwicklung des automatisierten Findens medizinischer Anomalien wird über vier Schritte implementiert: Erstens die Bereinigung technischer Artefakte, zweitens die Herzschlagerkennung und -segmentierung, drittens die Feature-Extraktion und viertens die Klassifikation. Mithilfe der nachfolgenden Methode (Code-Ausschnitt B.1) werden die ersten zwei Schritte auf dem MIT-BIH-Datensatz implementiert. Die DWT-Koeffizienten und die RR-Intervalle werden ebenfalls bereits schon als Feature pro Herzschlag extrahiert, was dem dritten Schritt zuzuordnen ist. Je nachdem ob binär oder über mehrere Klassen klassifiziert wird, weichen die einzelnen Vorverarbeitungsschritte ein wenig voneinander ab. Da sie jedoch grundsätzlich sehr ähnlich aufgebaut sind, wird hier nur eine mögliche Implementierung dargestellt.

B.1 QUELLCODE-AUSSCHNITTE

```
pathDf = DataFrame(z.get("pathDf"), sqlContext)
detectors = Detectors(360)

def discrete_wavelet_decomposition(wavelet_type, y_noise):
    DWTcoeffs = pywt.wavedec(y_noise,wavelet_type,mode='symmetric', level=8,
     axis=-1)
    DWTcoeffs[0] = np.zeros_like(DWTcoeffs[0])
    DWTcoeffs[-1] = np.zeros_like(DWTcoeffs[-1])
    DWTcoeffs[-2] = np.zeros_like(DWTcoeffs[-2])
    DWTcoeffs[-3] = np.zeros_like(DWTcoeffs[-3])
    return pywt.waverec(DWTcoeffs,wavelet_type,mode='symmetric',axis=-1)

udf1 = udf(lambda x:x[1:12],StringType())
udf2 = udf(lambda x:x[13:21],StringType())
udf3 = udf(lambda x:x[26:27],StringType())
udf4 = udf(lambda x:x[31:32],StringType())
udf5 = udf(lambda x:x[36:37],StringType())
udf6 = udf(lambda x:x[41:42],StringType())

classes = ['N', 'L', 'R', 'A', 'V', '/']
window_size = 160
window_size_left = 100
window_size_right = 200
maximum_counting = 10000
n_classes = len(classes)
count_classes = [0]*n_classes
mitbih_pdf = pd.DataFrame(columns=['csv_number', 'annotation_index', '
    Ableitung1_heartbeat', 'Ableitung2_heartbeat', 'label', 'AbleitungA', '
    AbleitungB', 'previous_RR', 'post_RR', 'local_RR', 'average_RR', 'DWTCoeffs1
    ', 'DWTCoeffs2'])

for row in pathDf.rdd.collect():
```

```
path = row["PathData"]
path_anno = row["PathAnnotation"]

data = spark.read.option("header",True).csv(path).repartition(1)
columns = data.columns

#data preparation for annotation data
file = spark.read.csv(path_anno).repartition(1)
header=file.first()[0]
anno = file.filter(~col("_c0").contains(header))

anno = anno.withColumn('Time',udf1('_c0')).withColumn('SampleNumber',trim(
udf2('_c0'))).withColumn('Type',udf3('_c0')).withColumn('Sub',udf4('_c0')).
withColumn('Chan',udf5('_c0')).withColumn('Num',udf6('_c0')).select("Time",
"SampleNumber", "Type", "Sub", "Chan", "Num")
anno = anno.withColumn("SampleNumber", anno["SampleNumber"].cast(IntegerType
()))

#Read data and rename
data = data.withColumnRenamed("\'sample #\'","index").withColumnRenamed(
columns[1],"Ableitung1").withColumnRenamed(columns[2],"Ableitung2")
#Cast Datatypes
data = data.withColumn("Ableitung1", col("Ableitung1").cast(IntegerType())).
withColumn("Ableitung2", col("Ableitung2").cast(IntegerType())).withColumn("
index", col("index").cast(IntegerType()))
data = data.toPandas()

#Get data as sorted numpy array
normalised_data_Ableitung1 = np.asarray(data.sort_values(by=['index'])["
Ableitung1"]).flatten()
normalised_data_Ableitung2 = np.asarray(data.sort_values(by=['index'])["
Ableitung2"]).flatten()
#Denoising data with dmey wavelet
clean_data_Ableitung1 = discrete_wavelet_decomposition("dmey",
normalised_data_Ableitung1)
clean_data_Ableitung2 = discrete_wavelet_decomposition("dmey",
normalised_data_Ableitung2)


r_peaks = detectors.two_average_detector(clean_data_Ableitung1)

#create new dataframe with arrays of heartbeats (II and V5) and label
#Extract heartbeat from clean data time series for each row in annotation
tabel
previous_RR_list = []
csv_number = path[-7:-4]
for row in anno.rdd.collect():
    pos = row["SampleNumber"]
    arrhythmia_type = row["Type"]

    if(arrhythmia_type in classes):
                arrhythmia_index = classes.index(arrhythmia_type)
                if count_classes[arrhythmia_index] > maximum_counting: #
avoid overfitting
                    pass
                else:
                    count_classes[arrhythmia_index] += 1
```

```
                    range_of_pos = np.arange(pos-30, pos+30)
                    matches = list(set(r_peaks) & set(range_of_pos))
                    if(len(matches) == 1):
                        if(window_size_left < matches[0] and matches[0] < (
len(clean_data_Ableitung1) - window_size_right)):
                            index_r_peak = r_peaks.index(matches[0])
                            if(index_r_peak > 0):
                                previous_RR = r_peaks[index_r_peak]-r_peaks[
index_r_peak-1]
                                previous_RR_list.append(previous_RR)
                            else:
                                previous_RR = None
                            if(index_r_peak+1 < len(r_peaks)):
                                post_RR = r_peaks[index_r_peak+1]-r_peaks[
index_r_peak]
                            else:
                                post_RR = None
                            if(len(previous_RR_list) == 0):
                                local_RR = None
                            elif(len(previous_RR_list)-10 < 0):
                                local_RR = mean(previous_RR_list[0:len(
previous_RR_list)])
                            else:
                                local_RR = mean(previous_RR_list[len(
previous_RR_list)-10:len(previous_RR_list)])
                            if(len(previous_RR_list) == 0):
                                average_RR = None
                            elif(len(previous_RR_list)-240 < 0):
                                average_RR = mean(previous_RR_list[0:len(
previous_RR_list)])
                            else:
                                average_RR = mean(previous_RR_list[len(
previous_RR_list)-240:len(previous_RR_list)])

                            Abl1 = clean_data_Ableitung1[matches[0]-
window_size_left+1:matches[0]+window_size_right].astype(np.float)
                            Abl2 = clean_data_Ableitung2[matches[0]-
window_size_left+1:matches[0]+window_size_right].astype(np.float)

                            y_up_Abl1 = Abl1 - np.min(Abl1)
                            y_up_Abl2 = Abl2 - np.min(Abl2)
                            y_max_Abl1 = np.max(y_up_Abl1)
                            y_max_Abl2 = np.max(y_up_Abl2)
                            y_norm_Abl1 = (y_up_Abl1/y_max_Abl1)
                            y_norm_Abl2 = (y_up_Abl2/y_max_Abl2)

                            diff = 320 - len(y_norm_Abl1)
                            zeros = np.repeat(0, diff)
                            newAbl1 = np.append(y_norm_Abl1, zeros)

                            diff2 = 320 - len(y_norm_Abl2)
                            zeros2 = np.repeat(0, diff2)
                            newAbl2 = np.append (y_norm_Abl2, zeros2)

                            DWTcoeffs1_ = pywt.wavedec(newAbl1,'db8',mode='
symmetric', level=4, axis=-1)
```

```
                                DWTcoeffs2_ = pywt.wavedec(newAbl2,'db8',mode='
    symmetric', level=4, axis=-1)

                                DWTcoeffs1 = np.hstack((DWTcoeffs1_[-3],
    DWTcoeffs1_[-4], DWTcoeffs1_[0]))
                                DWTcoeffs2 = np.hstack((DWTcoeffs2_[-3],
    DWTcoeffs2_[-4], DWTcoeffs2_[0]))

                                mitbih_pdf = mitbih_pdf.append({'csv_number':
    csv_number, 'annotation_index': pos, 'Ableitung1_heartbeat':newAbl1, '
    Ableitung2_heartbeat':newAbl2 , 'label': arrhythmia_type, 'AbleitungA':
    columns[1], 'AbleitungB': columns[2],'previous_RR': previous_RR, 'post_RR':
    post_RR, 'local_RR': local_RR, 'average_RR': average_RR, 'DWTCoeffs1':
    DWTcoeffs1, 'DWTCoeffs1': DWTcoeffs2}, ignore_index=True)

#convert to spark float array
mitbih_pdf.Ableitung1_heartbeat = mitbih_pdf.Ableitung1_heartbeat.map(lambda x:
    [float(e) for e in x])
mitbih_pdf.Ableitung2_heartbeat = mitbih_pdf.Ableitung2_heartbeat.map(lambda x:
    [float(e) for e in x])
mitbih_pdf.DWTCoeffs1 = mitbih_pdf.DWTCoeffs1.map(lambda x: [float(e) for e in x
    ])
mitbih_pdf.DWTCoeffs2 = mitbih_pdf.DWTCoeffs1.map(lambda x: [float(e) for e in x
    ])

#create spark dataframe
mitbih_df = spark.createDataFrame(mitbih_pdf).na.drop().withColumn("id",
    monotonically_increasing_id())
```

Listing B.1: Einlesen der MIT-BIH-Daten und Bereinigung der Zeitreihe sowie Herzschlagerkennung und -segmentierung.

Die Feature-Extraktion der unabhängigen Komponenten der Herzschläge und die Hauptkomponentenanalyse wird mithilfe des nachfolgenden Codes B.2 durchgeführt:

```
ica_abl1 = np.asarray(mitbih_df.select("Ableitung1_heartbeat").collect())
ica_abl2 = np.asarray(mitbih_df.select("Ableitung2_heartbeat").collect())
ica_abl1_norm = []
ica_abl2_norm = []
for i in ica_abl1:
    ica_abl1_norm.append(i.flatten())

for i in ica_abl2:
    ica_abl2_norm.append(i.flatten())

transformer = FastICA(n_components=14, random_state=0)
X_transformed_A = transformer.fit_transform(ica_abl1_norm)
X_transformed_B = transformer.fit_transform(ica_abl2_norm)


features_pca_abl1_coeffs1 = np.asarray(mitbih_df.select("DWTCoeffs1").collect())
features_pca_abl1_coeffs2 = np.asarray(mitbih_df.select("DWTCoeffs2").collect())

features_coeffs1_norm = []
for i in features_pca_abl1_coeffs1:
    features_coeffs1_norm.append(i.flatten())
```

```
features_coeffs2_norm = []
for i in features_pca_abl1_coeffs2:
    features_coeffs2_norm.append(i.flatten())

features_coeffs1_norm = np.asarray(features_coeffs1_norm)
features_coeffs2_norm = np.asarray(features_coeffs2_norm)

pca_matrix_A = np.concatenate((X_transformed_A,features_coeffs1_norm),axis=1)
pca_matrix_B = np.concatenate((X_transformed_B,features_coeffs2_norm),axis=1)


pca = PCA(n_components=19)
principalComponentsA = pca.fit_transform(pca_matrix_A)
principalComponentsB = pca.fit_transform(pca_matrix_B)

principalDf_A = pd.DataFrame(data = principalComponentsA
             , columns = ['A_pca_1', 'A_pca_2', 'A_pca_3', 'A_pca_4', 'A_pca_5',
     'A_pca_6', 'A_pca_7', 'A_pca_8', 'A_pca_9', 'A_pca_10', 'A_pca_11', '
    A_pca_12', 'A_pca_13', 'A_pca_14', 'A_pca_15', 'A_pca_16', 'A_pca_17', '
    A_pca_18', 'A_pca_19'])

principalDf_B = pd.DataFrame(data = principalComponentsB
             , columns = ['B_pca_1', 'B_pca_2', 'B_pca_3', 'B_pca_4', 'B_pca_5',
     'B_pca_6', 'B_pca_7', 'B_pca_8', 'B_pca_9', 'B_pca_10', 'B_pca_11', '
    B_pca_12', 'B_pca_13', 'B_pca_14', 'B_pca_15', 'B_pca_16', 'B_pca_17', '
    B_pca_18', 'B_pca_19'])


mitbih_pdf = mitbih_df.toPandas()
mitbih_pdf_incl_features = pd.concat([mitbih_pdf, principalDf_A, principalDf_B],
     axis=1)
```

Listing B.2: Feature-Extraktion der unabhängigen Komponenten und der Hauptkomponenten.

Die Implementierung des Klassifikationsmodells, welches die ausgewählten Parameter enthält, ist durch den Code-Ausschnitt B.3 dargestellt.

```
clf_rbf = SVC(kernel='rbf',class_weight='balanced', C=10, gamma='scale',
    shrinking=True)
clf_rbf.fit(X_train,Y_train)

train_score = clf_rbf.score(X_train, Y_train)
test_score = clf_rbf.score(X_test, Y_test)
print("The Train Score is {}".format(train_score))
print("The Test Score is {}".format(test_score))

Y_pred = clf_rbf.predict(X_test)
pd.crosstab(Y_test, Y_pred, rownames=['True'], colnames=['Predicted'], margins=
    True)
```

Listing B.3: Klassifikationsmodell als SVM und Auswertung der Klassifikation

B.2 ERGEBNISSE DER VERSCHIEDENEN KLASSIFIZIERER

Innerhalb des Prozesses zur Entwicklung einer Möglichkeit medizinische Anomalien automatisiert und zuverlässig zu finden, wurden verschiedene Klassifizierer implementiert. Die Ergebnisse der einzelnen Implementationen sind den nachfolgenden Tabellen B.1 und B.2 zu entnehmen.

Id	Typen	Methode	Ableitung	TP	FP	TN	FN
1	6	SVM	A	4.828	12	1.923	28
2	6	SVM	B	4.830	11	1.924	26
3	6	KNN	A	4.818	22	1.913	38
4	6	KNN	B	4.826	21	1.914	30
5	6	LDA	A	4.529	474	1.461	315
6	6	LDA	B	4.538	456	1.479	397
7	2	SVM	A	5275	26	1906	29

Tabelle B.1: Teil 1: Ergebnisse der Modelle zur Klassifikation von sechs Herzschlag-Typen in den Zeilen eins bis sechs und einer binären Klassifikation, bei der 16 verschiedene Herzschlag-Typen mit einbezogen wurden in der Zeile 7.

Id	Sensitivität	Falsch-Alarme	Accuracy
1	99,75%	0,62%	99,13%
2	99,77%%	0,57%	99,19%
3	99,54%	1,14%	98,66%
4	99,57%	1,09%	98,82%
5	90,53%	24,50%	82,14%
6	90,87%	23,57%	82,52%
7	99,45%	1,35%	99,23%

Tabelle B.2: Teil 2: Ergebnisse der Modelle zur Klassifikation von sechs Herzschlag-Typen in den Zeilen eins bis sechs und einer binären Klassifikation, bei der 16 verschiedene Herzschlag-Typen mit einbezogen wurden in der Zeile 7.

Die Abbildung B.1 zeigt die Ergebnisse verschiedener Paper zur Herzschlagklassifikation, zusammengestellt von Li et al. [37].

Table 4. Heartbeat classification performance on the MIT-BIH dataset.

Method	Year	Abnormal/Normal	Heartbeat Types	TP	FP	TN	FN	Sensitivity	False Alarm	Accuracy
Christov et al. [30]-morphology	2006	18,378/47,239	5	180,42	1604	45,635	336	98.17%	3.40%	97.04%
Christov et al. [30]-frequency	2006	18,378/47,239	5	17,590	1459	45,780	788	95.71%	3.09%	96.58%
Chazal et al. [31]-frequency	2006	4317/34,394	5	4108	1962	32,432	209	95.16%	5.70%	94.39%
Ubeyli et al. [83]	2009	269/90	4	268	2	88	2	99.26%	2.22%	99.89%
Llamedo et al. [38]	2010	5441/44,188	3	4752	2238	41,950	689	87.34%	5.06%	94.10%
Ye et al. [71]-rejection	2012	19,913/64,042	16	19,815	93	63,949	98	99.51%	0.15%	99.77%
Ye et al. [71]-bayesian	2012	20,745/65,264	16	20,557	286	64,978	188	99.09%	0.44%	99.45%
Zhang et al. [74]	2014	5653/44,011	4	5248	4869	39,142	405	92.84%	11.06%	89.38%
Thomas et al. [32]	2015	26,626/672,68	5	22,900	1300	65,968	3726	86.01%	1.93%	94.65%
Kiranyaz et al. [36]	2015	7366/42,191	5	6539	1228	40,963	827	88.77%	2.97%	95.85%
Rajesh et al. [33]	2017	8000/2000	5	7677	33	1967	323	95.96%	1.65%	96.44%
Sahoo et al. [75]	2017	807/244	4	798	5	239	9	98.88%	2.04%	98.67%

Abbildung B.1: Vergleich der Modell-Performance verschiedener Paper zur Herzschlagklassifikation aus Li et al. [37].