INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Diplomarbeit

Entwurf und Implementierung einer Software-Umgebung zur Analyse der Vorhersage epileptischer Anfälle aus Langzeit-EEG-Daten

Hinnerk Feldwisch genannt Drentrup

Aufgabensteller:	Prof. Dr. Hans Jürgen Ohlbach
Betreuer:	Prof. Dr. Jens Timmer
Abgabetermin:	4. Mai 2006

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 3. Mai 2006 _____

Unterschrift

Zusammenfassung

Für Epilepsiepatienten stellen die überraschend eintretenden Anfälle ein großes Bedrohungspotential dar. Verfahren zur Vorhersage epileptischer Anfälle könnten daher neuartige Behandlungsmöglichkeiten der mit den Anfällen einhergehenden Symptome bieten. Basierend auf verschiedenen Methoden u. a. der Nichtlinearen Dynamik, einem Teilgebiet der Physik, wurden zu diesem Zweck in den vergangenen Jahren Algorithmen entwickelt, um aus gemessenen Hirnströmen Informationen zu gewinnen, die eine Vorhersage und eine damit verbundene Intervention zur Verhinderung von Anfällen ermöglichen sollen [1].

Um die Aussagekraft der Vorhersagemethoden zu untersuchen, müssen diese systematisch angewandt und analysiert werden. Im Mittelpunkt dieser Diplomarbeit steht die Aufgabe, eine Software-Umgebung gemäß den spezifischen Anforderungen der Anfallsvorhersage zu entwerfen und zu implementieren. Ein wesentlicher Aspekt des realisierten Programms ist dessen flexibel und modular gestaltete Struktur. Diese erlaubt die Verwendung verschiedenster Parameter, über welche wichtige Details der Vorhersage gesteuert werden können. Neben der Anwendung bereits implementierter Vorhersagemethoden wurden auch Schnittstellen zur Einbindung neuer Verfahren bereitgestellt.

Zur Auswertung und statistischen Analyse der Vorhersagemethoden können Langzeitaufnahmen des Elektroenzephalogramms (EEG) von Epilepsiepatienten verwendet werden. Besonderer Wert wird auf eine robuste und nachvollziehbare Verarbeitung der Daten gelegt, für welche verschiedene *XML*-Formate definiert wurden. Durch die Entwicklung eines online-fähigen Vorhersagesystems wird die Durchführung von Untersuchungen ermöglicht, mit denen die Perspektiven zukünftiger praktischer Anwendungen der Anfallsvorhersage analysiert werden können.

Abstract

The unexpected occurrence of epileptic seizures poses a dangerous threat to patients. A reliable prediction of epileptic seizures could offer novel treatment strategies for the disease epilepsy. To this aim, several algorithms have been developed during the past years, which are based on methods originating from Nonlinear Dynamics. Specific information is extracted from brain currents to enable a prediction and subsequent prevention of seizures [1].

To evaluate seizure prediction methods they have to be applied and analyzed systematically. The central challenge of this diploma thesis is the design and implementation of a software framework for this purpose, in accordance with the specific requirements of seizure prediction. The developed software is characterized by its flexibility and modular structure. It enables usage of various parameters to control important prediction details. In addition to some already implemented techniques, appropriate interfaces were designed for the integration of novel seizure prediction methods.

Long-term continuous electroencephalographic measurements of epilepsy patients can be utilized by the software for the assessment and statistical evaluation of seizure prediction methods. Robust data processing is of particular importance. The developed online prediction system allows manifold experiments to examine future applications of seizure prediction.

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Motivation 1
1.2 Herausforderungen und Ziele dieser Diplomarbeit 2
1.3 Gliederung und Aufbau 3

2 Grundlagen **5**

2.1 Epilepsie und epileptische Anfälle 5
2.1.1 Behandlungsmöglichkeiten 6
2.2 Erhobene Daten 7
2.2.1 EEG-Aufzeichnungen 7
2.2.2 Erhobene Zusatzinformationen 8
2.3 Ermittlung charakteristischer Zeitreihen aus dem EEG 10
2.3.1 Phasensynchronisationsindex 10
2.3.2 Dynamical Similarity Index 12
2.4 Die „Seizure Prediction Characteristic“ 12
2.5 Zusammenfassung 15

3 Konzeptioneller Entwurf der Software-Umgebung **17**

3.1 Zentrale Anforderungen 17
3.1.1 Flexibler Einsatz und Erweiterbarkeit 17
3.1.2 Benötigte Typen der Vorhersagemethoden 18
3.1.3 Kombination von Vorhersagemethoden 19
3.1.4 Online-Vorhersage 20
3.2 Der Analyse-Prozess 20
3.3 Entwurf des Analyseprogrammes „SeizureAnalyzer“ 23
3.3.1 Objektorientierte Programmierung 23
3.3.2 Übersicht über die Programmstruktur 24
3.3.3 Datenverarbeitung 26
3.3.4 Durchführung der Analyse für Daten eines Patienten 28
3.3.5 Entwurf von Vorhersagemethoden 28
3.4 Zusammenfassung 33

4 Implementierung des „SeizureAnalyzer“ 35
4.1 Überblick . . . 35
4.2 Konfiguration . . . 37
4.2.1 Die „Extensible Markup Language“ *XML* . . . 37
4.2.2 *XML* als Konfigurations-Format . . . 39
4.3 Datenverarbeitung . . . 42
4.4 Module zur Vorhersage-Analyse . . . 43
4.4.1 Offline-Vorhersage . . . 43
4.4.2 Online-Vorhersage . . . 45
4.4.3 Kombination von Vorhersagemethoden . . . 47
4.5 Bereitstellung der Ergebnisse für weitere Auswertungen . . . 50
4.6 Zusammenfassung . . . 52

5 Auswertungen und Ergebnisse 53
5.1 Beschreibung der ausgewerteten EEG-Daten . . . 53
5.2 Visualisierung der erzeugten Alarme . . . 55
5.3 Abhängigkeit der Sensitivität von den Vorhersage-Zeitfenstern . . . 57
5.4 Kombination zweier Vorhersagemethoden . . . 58
5.5 Lern- und Test-Ergebnisse . . . 60
5.6 Zusammenfassung . . . 64

6 Zusammenfassung und Ausblick 65

Anhang iii

A Mathematische Herleitung der verwendeten Vorhersagemethoden iii
A.1 Phasensynchronisationsindex . . . iii
A.2 Dynamical Similarity Index . . . v
A.3 Random Predictor . . . vi

B Verwendete *XML*-Formate ix
B.1 Konfiguration des Analyse-Projektes . . . ix
B.2 Konfiguration der Patienten-Informationen . . . x
B.3 Konfiguration der Datenquellen . . . x
B.4 Ausgabeformat für die Vorhersage-Ergebnisse . . . xi

C Quellcode des PredictionSystem xiii

D Technische Details der Implementation xvii

Literaturverzeichnis xvii

Kapitel 1

Einleitung

1.1 Motivation

Die Gefahr, die von der häufigsten chronischen Erkrankung des Nervensystems, der Epilepsie, ausgeht, liegt in den überraschend einsetzenden Anfällen. Diese treten oft ohne jegliche Vorwarnung auf und können durch Stürze und Krämpfe mit einhergehendem Bewußtseinsverlust zu schweren Verletzungen, zu Atemstillstand und weiteren lebensbedrohlichen Situationen führen. Auch wenn die frühere Stigmatisierung der Epilepsie als „göttliche Krankheit“ und „dämonische Besessenheit“ in westlichen Ländern inzwischen vernachlässigt werden kann, sind die Patienten weiterhin in ihrem alltäglichen Leben teilweise sehr stark eingeschränkt. Neben den direkten Krankheitssymptomen bestehen weitere Gründe für Beeinträchtigungen in den Medikamentennebenwirkungen, sowie in den psychosozialen Auswirkungen der unkontrolliert auftretenden Anfälle.

Die Vorhersage epileptischer Anfälle ist daher eine der wichtigsten Forschungsaufgaben zur Behandlung der Epilepsie [2]. Wäre es möglich, rechtzeitig und mit ausreichend hoher Genauigkeit vor den einzelnen Anfällen Alarme auszulösen, könnte zeitlich gezielt medizinisch interveniert und die akute Bedrohung aufgehoben werden.

Den Anfällen gehen jedoch nur selten spezifische Hinweise voraus [3]. Im Normalfall erleiden die Patienten ihre Anfälle unvermittelt. In den vergangenen Jahren wurden erste Methoden vorgestellt, um mit Hilfe verschiedener Algorithmen Anfälle automatisch vorherzusagen [1, 4]. Hierzu werden aus EEG-Aufzeichnungen Kenngrößen der Nichtlinearen Dynamik ermittelt, durch welche typische Veränderungen vor Anfällen festgestellt werden konnten. Es sind systematische und statistische Auswertungen der Ergebnisse nötig, um profunde Aussagen über die Vorhersagekraft und Signifikanz

der Methoden machen zu können. Zur Bewertung ist hierbei neben der Sensitivität, d. h. dem Anteil der korrekt vorhergesagten Anfälle, auch die Spezifität sehr wichtig, welche mit der Rate der Fehlalarme identifiziert werden kann. Letztere sollte möglichst klein sein, damit die Vorhersagemethode für Patienten und Ärzte nützlich und tolerabel ist.

1.2 Herausforderungen und Ziele dieser Diplomarbeit

Zur Analyse der Vorhersage epileptischer Anfälle werden zuverlässige und fehlertolerant implementierte Software-Komponenten benötigt, um eine automatisierte Anwendung der Vorhersagemethoden durchführen zu können. Ziel dieser Diplomarbeit ist es dementsprechend, eine Software-Umgebung zu entwerfen und zu realisieren, über welche die Optimierung und Auswertung von Vorhersagemethoden durchgeführt werden kann.

Genauso wie die Möglichkeiten zur Behandlung der Krankheit Epilepsie variieren auch die Möglichkeiten zur Vorhersage ihrer Anfälle stark von Patient zu Patient. Daher müssen die Vorhersagemethoden für jeden Patienten einzeln „trainiert“ werden können. Sowohl diese individuelle Optimierung als auch die Überprüfung der Ergebnisse sollte mit Hilfe möglichst langer EEG-Aufzeichnungen erfolgen. Während früher kaum mehr als ein- oder zweitägige Aufzeichnungen mit wenigen Anfällen zur Verfügung standen [4], wurden in den letzten Jahren vermehrt Langzeit-Aufzeichnungen über mehrere Tage vorgenommen, welche die Untersuchung interessanter Fragestellungen ermöglichen. Gleichzeitig führt es zu neuen Herausforderungen für die Verarbeitung und Auswertung der Daten. Diese müssen kontinuierlich und möglichst in Echtzeit durchgeführt werden, um einer zukünftigen, realen Anwendung der Vorhersagemethoden nahe zu kommen.

Durch den vielseitigen Einsatz der Software bedingt, ergibt sich als zentrale Anforderung ein hohes Maß an Flexibilität und Erweiterbarkeit. Verbesserte Arten der Anfallsvorhersage sowie z. B. zusätzliche Verfahren zur statistischen Evaluierung derselben müssen mit möglichst geringem Aufwand integriert werden können, damit die Software-Umgebung zur Erforschung neuartiger Ansätze eingesetzt werden kann.

1.3 Gliederung und Aufbau

Bei der Entwicklung des Software-Projektes dieser Diplomarbeit ist der Bezug zu dessen Anwendungsgebiet essentiell. Die spezifischen Rahmenbedingungen der Vorhersage epileptischer Anfälle müssen präzise erfasst sein, um für den derzeitigen Einsatz zu Forschungszwecken valide Ergebnisse erzielen zu können. Daher werden im Kapitel 2 nach einer kurzen Einführung der Epilepsie und der epileptischen Anfälle die verwendeten EEG-Daten vorgestellt, und ein Überblick über zwei Methoden der Nichtlinearen Dynamik gegeben, mit deren Hilfe charakteristische Zeitreihen aus den EEG-Daten gewonnen werden. Ferner werden die theoretischen Aspekte der Anfallsvorhersage erörtert.

Ausgehend von den Anforderungen an die Software-Umgebung werden in Kapitel 3 der allgemeine Prozess der Analyse von Vorhersage-Systemen ausgearbeitet, sowie der gewählte Entwurf für die zentrale Software-Komponente „SeizureAnalyzer“ dargelegt. Hierbei werden auch die verwendeten Entwurfstechniken vorgestellt.

Im Kapitel 4 wird die durchgeführte Implementierung des Programmes behandelt. Dies beinhaltet neben einer Erörterung der verwendeten Dokumentenformate und der Module zur Datenverarbeitung eine detaillierte Beschreibung der implementierten Vorhersagemethoden. Das Kapitel 5 dient der Darstellung erster Ergebnisse, welche anhand des Programms „SeizureAnalyzer“ im Rahmen dieser Diplomarbeit ermittelt wurden.

Kapitel 2

Grundlagen

2.1 Epilepsie und epileptische Anfälle

Die Epilepsie (von griechisch $\varepsilon\pi\iota\lambda\eta\psi\iota\alpha$ „Anfassen“, „Anfall“) ist eine weit verbreitete neurologische Erkrankung, die schon vom griechischen Arzt Hippokrates um 400 vor Christus korrekt als Funktionsstörung des Gehirns erkannt wurde. Gekennzeichnet ist sie durch Anfallserscheinungen, die durch synchrone Entladungen von Nervenzellen hervorgerufen werden. Mehr als fünf Prozent aller Menschen erleiden im Laufe des Lebens mindestens einen Anfall, doch spricht man von einer Epilepsie erst dann, wenn zusätzlich eine anhaltende Prädisposition für Anfälle vorliegt [5]. Dies betrifft in Deutschland 600.000 bis 800.000 Menschen, also ca. 0.8% der Bevölkerung – jährlich kommen ungefähr 30.000 neue Fälle hinzu [6].

Epileptische Anfälle können vielerlei Symptome hervorrufen. Vergleichsweise harmlos sind z. B. die speziell im Kindesalter auftretenden Absencen von 5 bis 30 Sekunden Länge, die nur durch Geistesabwesenheit und starren Gesichtsausdruck auffallen. Die relativ bekannten großen Krampfanfälle (französisch *Grand Mal* – großes Übel) werden meist durch plötzlichen Bewußtseinsverlust eingeleitet, gefolgt von einem schweren Krampfzustand der gesamten Körpermuskulatur, sowie von Zuckungen der Extremitäten und des Gesichtes. Falls mehrere Anfälle so dicht aufeinander folgen, dass die Patienten sich zwischenzeitlich nicht mehr erholen können, spricht man von einem *Status epilepticus*, der lebensbedrohlich ist und akut behandelt werden muss.

Charakterisiert sind epileptische Anfälle durch synchrone Entladungen benachbarter Nervenzellen im Gehirn, welche zu starker rhythmischer Aktivität führen. Diese breitet sich auch auf andere Bereiche aus, was zu „generalisierten“ Anfällen führen kann, bei denen das gesamte Gehirn Anfallsaktivität zeigt. Während der Anfälle kann das betroffene Nervengewebe seine normale Funktion nicht ausüben, so dass die An-

fälle zu Kontroll- und Bewußtseinsverlust führen können. Die in Folge auftretender Stürze oder Krämpfe entstehenden Verletzungen stellen die grundlegende Gefahr der Epilepsie dar, denn nur in wenigen Fällen spürt der Patient selber einen drohenden Anfall. Daher hat sich die *Amerikanische Gesellschaft für Epilepsie* als wichtigstes Forschungsziel die „Vorhersage, Früherkennung und Blockade von epileptischen Anfällen“ gesetzt [7].

2.1.1 Behandlungsmöglichkeiten

Die derzeit wichtigste Möglichkeit zur Behandlung der Epilepsie liegt in der Therapie durch eine Gruppe spezieller Medikamente, den Antiepileptika. Diese hemmen im Allgemeinen die Erregbarkeit der Neurone, so dass die epileptischen Anfälle symptomatisch unterdrückt werden. Bei ca. 50% aller neu diagnostizierten Epilepsiekranken kann die Anfallsfrequenz durch eine Dauertherapie mittels eines entsprechenden Antiepileptikums ausreichend verringert werden [8]. Jedoch kann für ca. 30% der Patienten auch durch eine individuell angepaßte Kombination mehrerer Medikamente keine Anfallsfreiheit erreicht werden. Auch können nicht in jedem Fall die teils starken Nebenwirkungen der eingesetzten Medikamente toleriert werden.

Als weitere Methode zur Behandlung spezieller Epilepsien haben sich chirurgische Eingriffe etabliert, bei denen Nervengewebe entfernt wird, in welchem die Anfälle entstehen. Die Operationen können durchgeführt werden, falls im Rahmen vorhergehender diagnostischer Abklärung ein eingegrenzter Fokus gefunden wird, von welchem sich die Anfallsaktivität ausbreitet. Dies erfolgt u. a. über die Auswertung von EEG-Aufzeichnungen. Auch wird mit verschiedenen Tests z. B. durch Elektrostimulationen abgeklärt, ob infolge des Eingriffes mit bleibenden kognitiven Beeinträchtigungen zu rechnen ist. Wenn diese umfangreiche Diagnostik positiv ausfällt, kann bei ca. 60% der operierten Patienten eine substanzielle Reduzierung der Anfallshäufigkeit erreicht werden, die bei bis zu 40% auch zur völligen Anfallsfreiheit führt [9].

Speziell für jene Patienten, denen weder durch langfristige, medikamentöse Therapien noch mit einem chirurgischen Eingriff geholfen werden kann, könnte sich durch eine Unterdrückung ihrer Anfälle mit Hilfe akuter Interventionen die Lebensqualität entscheidend verbessern. Einige der Antiepileptika sind sehr gut zur Akuttherapie geeignet, doch aufgrund fehlender Alarmmechanismen können sie derzeit nur eingesetzt werden, um in Notfällen wie bei einem „Status epilepticus“ einen längeren Anfall zu beenden.

Schon seit der Antike wurde versucht, durch Elektrostimulationen Anfälle zu behandeln [10]. Im Allgemeinen soll hierbei durch Stromstöße die Anfallsaktivität unter-

brochen werden, um das Gehirn „zurückzusetzen“. Die Stimulation des Vagusnerven ist inzwischen als Therapie etabliert, während die so genannte Tiefenhirnstimulation, die sich bei Morbus Parkinson als sehr erfolgreich erwiesen hat, in Bezug auf die Anwendung für Epilepsie noch weiter erforscht werden muss [10, 11].

2.2 Erhobene Daten

Wie sich herausgestellt hat, können mittels Methoden der Nichtlinearen Dynamik Voranfallszustände anhand gemessener Hirnströme detektiert werden [12, 13]. Daher besteht die Hoffnung, dass entsprechende Auswertungen auch Vorhersagen der Anfälle erlauben. Im Rahmen der derzeitigen Forschung kann dabei auf routinemäßig vorgenommene EEG-Ableitungen zurückgegriffen werden, welche zum Zweck der prächirurgischen Diagnostik in einigen Epilepsiezentren aufgezeichnet werden.

2.2.1 EEG-Aufzeichnungen

Elektrische Ströme haben eine zentrale Rolle beim Signalaustausch der Nervenzellen. Beim Hirnstrombild bzw. Elektroenzephalogramm (EEG) werden über Elektroden Gehirnströme registriert. Hierbei wird die zeitliche Modulation der elektrischen Potenziale großer Nervenzellverbände (Größenordnung $10^6 - 10^9$ Neurone) gemessen [14]. Zur Diagnose verschiedenster neurologischer Erkrankungen hat sich das EEG seit den ersten Messungen im Jahre 1924 weitgehend etabliert. Früh wurde entdeckt, dass Epilepsiekranke typische Veränderungen in ihrer hirnelektrischen Aktivität zeigen [15]. Speziell während der Anfälle sind sehr deutliche Muster mit großer Amplitude zu beobachten.

Häufig angewandt werden EEG-Registrierungen mit Elektroden an der Kopfoberfläche, mit welchen in grober räumlicher Auflösung die elektrische Aktivität des Gehirns gemessen werden kann. Um bei der prächirurgischen Diagnostik jedoch eine möglichst genaue Lokalisierung des epileptischen Fokus zu ermöglichen, werden Elektroden nicht auf der Kopfoberfläche angebracht, sondern intrakraniell (gr. *kranίon* – Schädel) implantiert – also direkt im Gehirn. Durch den direkten Kontakt mit dem Nervengewebe wird außerdem ein hohes Signal-Rausch-Verhältnis erzielt. In Abbildung 2.1 sind die üblicherweise benutzten Elektroden und exemplarische Implantationsorte dargestellt. Es werden sowohl Ableitungen an tiefen Regionen im Gehirn vorgenommen, als auch an der Hirnrinde, dem Cortex.

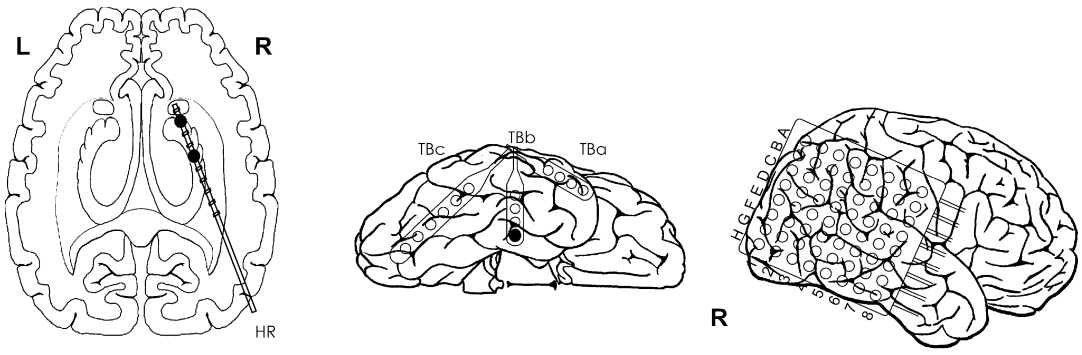


Abbildung 2.1: Verwendete Elektroden: Tiefenelektroden (links) mit Kontakten in tiefen Hirnstrukturen, und Streifenelektroden (mitte) sowie Gitterelektroden (rechts) mit Oberflächenkontakten.

In Abbildung 2.2 ist beispielhaft eine EEG-Registrierung von einer Minute Länge dargestellt, die einen Anfallsbeginn enthält. In dieser „bipolaren“ Darstellung sind die Spannungsdifferenzen von jeweils zwei Kontakten aufgetragen, anhand derer auf lokale elektrische Aktivitäten geschlossen werden kann. Im Beispiel wurden eine Gitterelektrode, vier Streifenelektroden und eine Tiefenelektrode verwendet. Nach ca. 32 Sekunden ist ein typischer Anstieg der Spannungsdifferenz zwischen den Kontakten HL_3 und HL_4 der Tiefenelektrode sichtbar, die der Depolarisation von Nervenzellen entspricht, welche in der Nähe dieser Elektroden liegen. Anschließend kommt es zu hochfrequenten Wellen mit großen Amplituden, welche sich auch auf andere Elektroden ausbreitet. Typisch ist ebenfalls die z. B. auf den Kanälen G_C3-G_C6 beobachtbare niedrigfrequente Aktivität mit rhythmischen Spikes. Im Gegensatz zu dieser deutlich sichtbaren Aktivität während der Anfälle können im Allgemeinen jedoch keine direkt erkennbaren Muster im EEG gefunden werden, welche systematisch den Anfällen vorausgehen.

Wie in diesem Falle erlaubt die Aufzeichnung des Elektroenzephalogramms von Epilepsiepatienten häufig die Lokalisierung des Hirnareals, in welchem die epileptischen Anfälle generiert werden. Die EEG-Messungen werden im Regelfall so lange durchgeführt, bis der epileptische Fokus gefunden ist, bzw. bis entschieden werden muss, dass keine ausreichenden Aussichten für einen erfolgreichen Eingriff gegeben sind.

2.2.2 Erhobene Zusatzinformationen

Schon während der Aufzeichnungen werden über so genannte „Marker“ wichtige Ereignisse mit ihren Zeitpunkten festgehalten (in Abbildung 2.2 blau eingezeichnet). Die

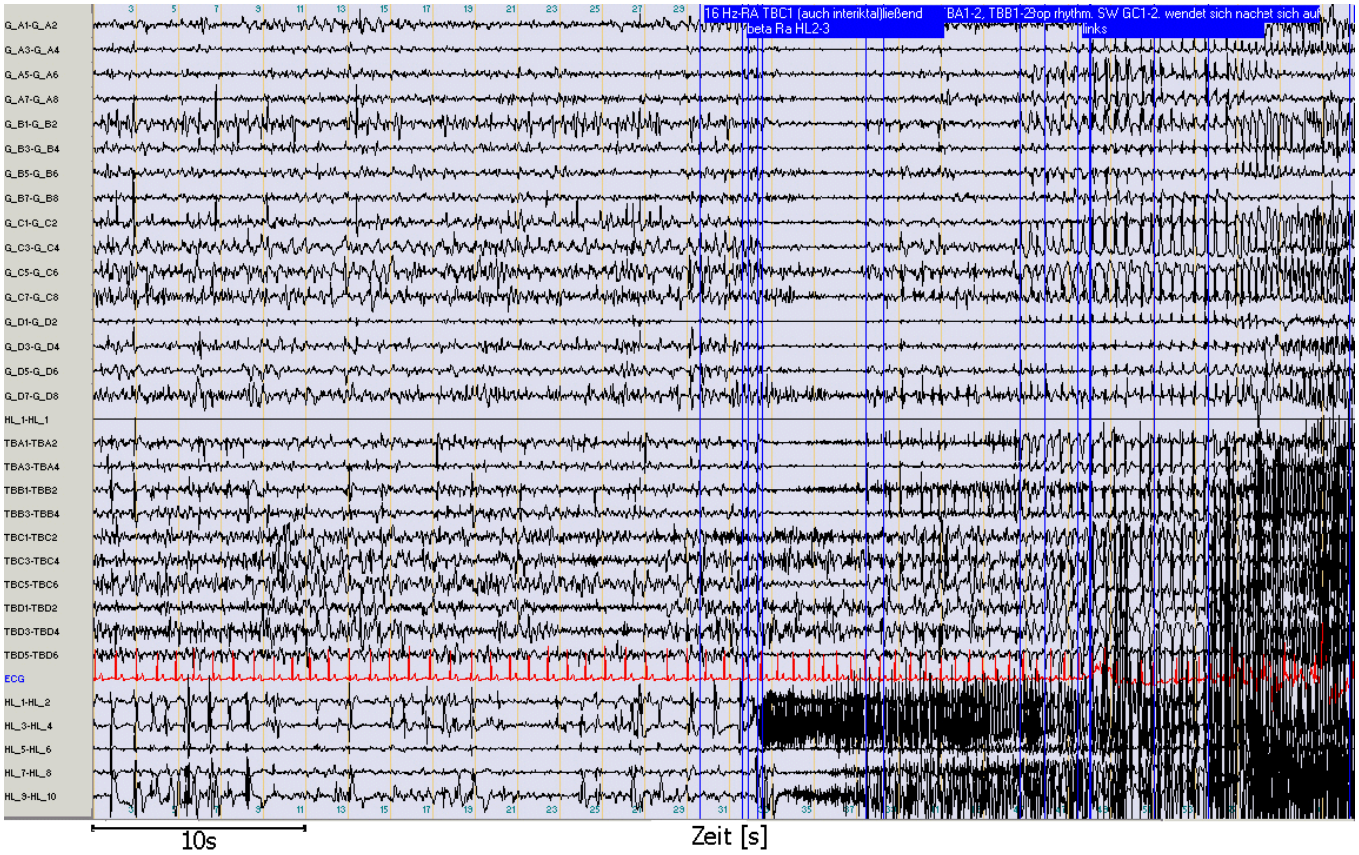


Abbildung 2.2: Ein-minütiger Ausschnitt einer EEG-Aufzeichnung während eines Anfallbeginns. Gemessen wurde mit einer 8×8 Gitterelektrode (G_A1 bis G_D8), vier Streifenelektroden (TBA1 bis TBD6) und einer Tiefenelektrode (HL_1 bis HL_10). Dargestellt sind die Spannungsdifferenzen zwischen je zwei Kontakten (G_A1-G_A2, etc.). Zusätzlich wurde mit dem Elektrokardiogramm (*ECG* in rot) die elektrische Herzaktivität aufgezeichnet. In blau mit vertikalen Linien markiert sind die Zeitpunkte klinisch wichtiger Ereignisse (siehe Abschnitt 2.2.2).

für die Analyse der Anfallsvorhersage wichtigsten Informationen sind die Start- und Endzeitpunkte der Anfälle, welche bei den Patienten im Aufnahmezeitraum eingetreten sind. Diese werden von einem erfahrenen Epileptologen anhand des Elektroenzephalogramms und bestimmter klinischer Symptome festgelegt. Mit Hilfe der Marker werden außerdem die Zeitpunkte von Elektrostimulationen und Veränderungen in der Medikation notiert, sowie weitere, klinisch interessante Ereignisse.

Bei den Patienten, deren EEG-Aufzeichnungen auch für die Anfallsvorhersage verwendet werden, konnte im Laufe dieser prächirurgischen Diagnostik der epileptische Fokus bestimmt werden. Da außerdem die exakte Lage der Elektroden zu rekonstruieren ist [16], kann auch ermittelt werden, ob die einzelnen Kontakte der Elektroden im Gebiet des Fokus liegen. All diese Informationen ermöglichen bei später entdeckten Auffälligkeiten, unter anderem in den EEG-Daten, eine Nachforschung ihrer Ursachen.

2.3 Ermittlung charakteristischer Zeitreihen aus dem EEG

Die traditionelle Inspektion und Analyse vom EEG oder von anderen physiologischen Daten ergab keine ausreichenden Hinweise auf nahende epileptische Anfälle. Für verschiedene Methoden der Zeitreihenanalyse wurden in den vergangenen Jahren erste Erfolge für die Erkennung von Veränderungen in der Dynamik von EEG-Signalen vor Anfällen berichtet [1, 17]. Als zwei erfolgversprechende Methoden wurden im Rahmen dieser Diplomarbeit der „Phasensynchronisationsindex“ [18, 19] und der „Dynamical Similarity Index“ [20, 21] verwendet.

Für beide Methoden wird das EEG als Zeitreihe aufgefaßt, d. h. als zeitlich diskrete Folge von einzelnen Datenpunkten. Es wird jeweils ein Fenster fester Länge analysiert, für welches ein charakteristischer Wert berechnet wird, wie es in Abbildung 2.3 dargestellt ist. Die Fenster werden dabei in jedem Schritt überlappend und um ein Zeitfenster Δt verschoben angesetzt. Das Ergebnis ist eine neue Zeitreihe, das „Feature“, welches zum Zwecke der Anfallsvorhersage verwendet werden kann.

2.3.1 Phasensynchronisationsindex

Aufgrund der weit verzweigten Interaktionen von Nervenzellen können Kopplungseffekte auch in den gemessenen EEG-Signalen beobachtet werden. Eine Methode zur

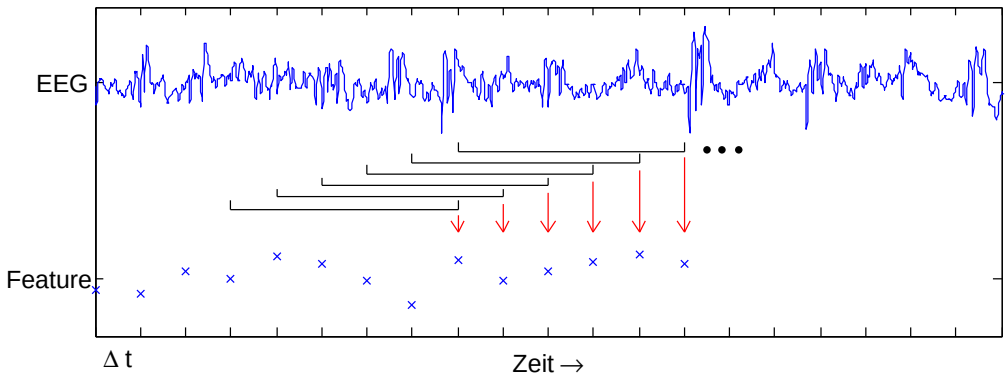


Abbildung 2.3: Schema der Berechnung einer charakteristischen Zeitreihe („Feature“) aus EEG-Daten. Iterativ werden für Fenster fester Länge die EEG-Daten analysiert, für die sich als Ergebnis der Analyse ein neuer Datenpunkt ergibt. Die Fenster werden dabei jeweils um die Zeitspanne Δt versetzt, welche dann auch dem Abstand der berechneten Datenpunkte entspricht.

quantitativen Beschreibung dieser Kopplung ist der „Phasensynchronisationsindex“, welcher die Synchronisation zweier schwingender Signale misst. Dabei werden nur die Phasen der Signale betrachtet, d. h. die instantanen Schwingungszustände, die als Phasenwinkel angegeben werden können.

Zur Berechnung des Indexes werden die Daten von beiden Kanälen fensterweise eingelesen, und jeweils die Phaseninformationen rekonstruiert. Falls die Phasen beider Signale im zeitlichen Mittel um einen festen Phasenwinkel gegeneinander verschoben sind, ergibt sich in diesem gekoppelten Fall für den Phasensynchronisationsindex der Wert 1, für völlig unkorrelierte Phasen der Wert 0. Eine mathematische Herleitung des Phasensynchronisationsindexes erfolgt in Anhang A.1.

Zur Ermittlung des Indexes wurde bereits ein Verfahren vorgeschlagen [22], bei welchem über ein so genanntes Zellulares Neuronales Netz in spezieller Hardware eine sehr schnelle Abschätzung des ansonsten aufwändig zu berechnenden Indexes durchgeführt werden kann. Insbesondere für den Einsatz in einem integrierten Vorhersagesystem ist dies ein vielversprechender Ansatz, da es die Abschätzung in Echtzeit und mit sehr niedrigem Energieverbrauch erlauben würde.

2.3.2 Dynamical Similarity Index

Im Gegensatz zum Phasensynchronisationsindex ist der „Dynamical Similarity Index“ ein univariates Verfahren, d. h. es wird nur das Signal eines EEG-Kanals betrachtet [20]. Die zugrunde liegende Idee besteht in dem Vergleich des jeweils aktuellen Datenfenster mit einem vorher ausgewählten Referenzfenster, welches weit von Anfällen entfernt liegt. Dabei werden die Daten nicht direkt verglichen, sondern anhand ihrer dynamischen Eigenschaften. Falls das aktuell vorliegende Datenfenster in seiner Dynamik dem Referenzfenster sehr ähnlich ist, liegt der Wert des normierten Similarity Index nahe 1, andernfalls eher bei 0. Für jeden EEG-Kanal, für den der Similarity Index ermittelt wird, ergibt sich eine neue Zeitreihe bzw. Feature. Für eine detailliertere Beschreibung des Similarity Index siehe Anhang A.2.

2.4 Die „Seizure Prediction Characteristic“

Die im vorherigen Abschnitt vorgestellten Indices zeigen für einige Patienten charakteristische Veränderungen in der präiktalen Phase, d. h. im Zeitraum vor den Anfällen. Zur Detektion dieser Veränderungen werden Schwellwerte benutzt, bei deren Überschreitungen Alarme[1] ausgelöst werden. In Abbildung 2.4 ist dies für den Phasensynchronisationsindex gezeigt, der aus den beiden abgebildeten EEG-Kanälen berechnet wurde. Vor dem markierten Anfall überschreitet der Index die eingezeichnete Schwelle a. Zur Optimierung des Vorhersagesystems muss für jeden Patienten ein bester Schwellwert ermittelt werden, bei deren Über- bzw. Unterschreitung Alarme ausgelöst werden, welche die Anfälle ankündigen. Dabei muss die Anzahl der korrekt vorhergesagten Anfälle maximiert, die Anzahl der Fehlalarme minimiert werden. Außerdem ist zu untersuchen, wie groß die Vorhersage-Zeitfenster sein müssen oder sein dürfen, um nützliche und signifikante Vorhersagen treffen zu können. Hierbei wurde im Rahmen der Diplomarbeit auf die „Seizure Prediction Characteristic“ (engl. *Seizure* – Anfall) zurückgegriffen, mit der eine statistische Bewertung der Vorhersagemethoden möglich ist [21, 23, 24].

Das Ziel der Vorhersagen ist es, vor den Anfällen eine Intervention zu ermöglichen, um die Anfälle unterdrücken zu können. Die Vorhersagemethode muss daher nach Auslösung eines Alarmes eine Interventionszeit beinhalten, damit z. B. Notfallmedikamente genommen werden und ihre Wirkung entfalten können. Während dieses „Seizure Prediction Horizon“ (SPH) darf kein Anfall eintreten, damit die Intervention ungestört durchgeführt werden kann.

[1] Alarm: von ital. *all'arme* – „Zu den Waffen“

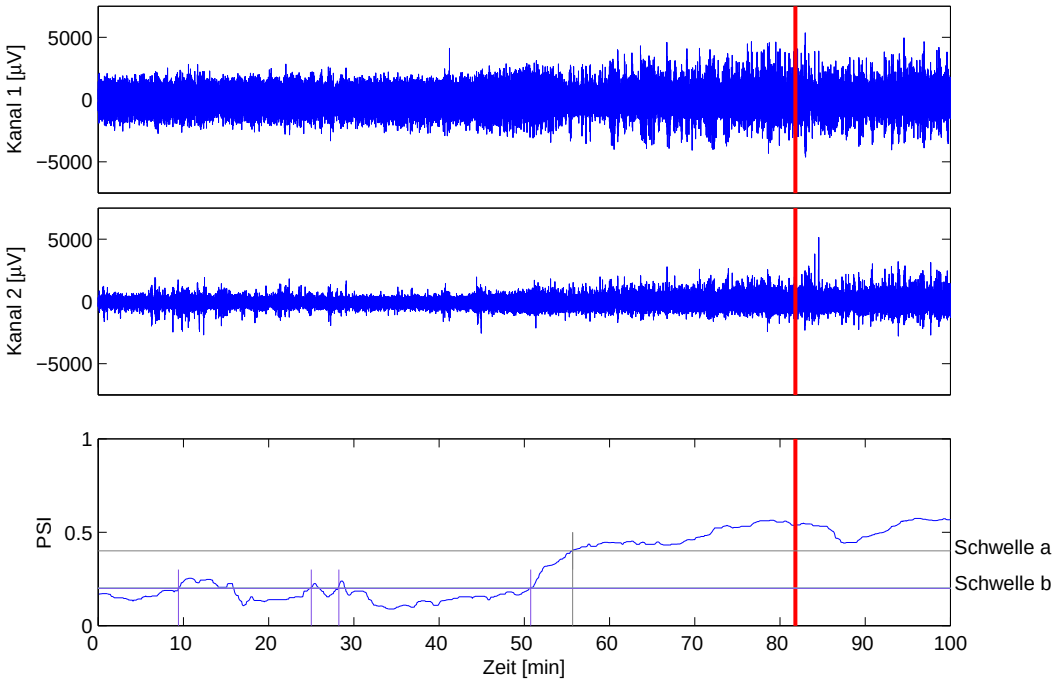


Abbildung 2.4: Beispiel für die Auswertung des Phasensynchronisationsindex für die Vorhersage: von den oben gezeigten Signalen zweier EEG-Kanäle wurde die Zeitreihe des Phasensynchronisationsindex *PSI* berechnet. Mit der roten vertikalen Linie ist ein Anfall markiert, der während des gezeigten Zeitfensters eintrat. Zur Detektion von Anstiegen des Indexes werden Schwellwerte verwendet, von denen beispielhaft zwei eingezeichnet sind. In diesem Fall kann durch die Auslösung eines Alarmes bei Überschreitung der Schwelle a ca. 27 Minuten vor dem Anfall gewarnt werden.

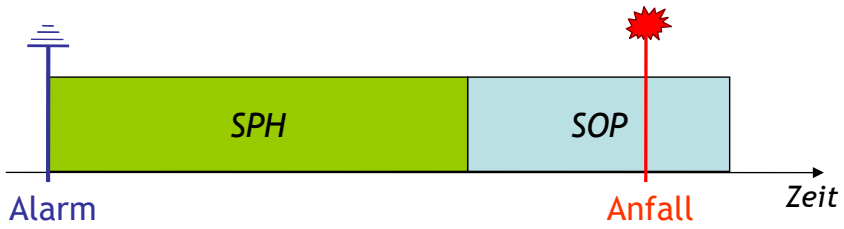


Abbildung 2.5: Vorhersagefenster nach einem ausgelösten Alarm. Während des „Seizure Prediction Horizon“ *SPH* soll eine Intervention erfolgen können, um den Anfall zu unterdrücken. Daher darf während dieses Zeitfensters kein Anfall eintreten. Vorhergesagt wird der Anfall für die anschließende „Seizure Occurence Period“ *SOP*.

Das Zeitfenster, innerhalb dessen anschließend der Anfall erwartet wird, ist die „Seizure Occurence Period“ (*SOP*) – siehe auch Abbildung 2.5. Bei einer echten Anwendung, bei der die Daten in Echtzeit ausgewertet und Alarme ausgelöst werden, muss während dieses Zeitraumes die Intervention wirken, um den vorhergesagten Anfall zu unterdrücken. Jedoch können bisher nur bereits vorhandene Daten ausgewertet werden. Da für diese aber bekannt ist, zu welchen Zeitpunkten Anfälle eingetreten sind, ist es möglich, die Vorhersagemethoden sowohl für die Vorhersage dieser Anfälle zu trainieren, als auch sie zu testen.

Im Falle der retrospektiven Auswertung der Daten wird ein Alarm dann als korrekt definiert, wenn der dem Alarm folgende Zeitraum der Länge *SPH* anfallsfrei ist, und anschließend während des Zeitfensters *SOP* der erwartete Anfall eintritt. Falls sich während des *SOP*-Fensters kein Anfall ereignet, so ist der Alarm als Fehlalarm einzuordnen.

Ein zentrales Kriterium zur Bewertung einer Vorhersagemethode ist seine Sensitivität, d. h. der Anteil der korrekt vorhergesagten Anfälle an der Gesamtzahl aller eingetretenen Anfälle. Zur sinnvollen Bewertung der Vorhersageleistung muss die Sensitivität jedoch zusammen mit der erzielten Spezifität betrachtet werden, die als Anzahl der falsch positiven Alarme (pro Zeiteinheit) definiert wird. Insbesondere bei aufwändigen Interventionen wird diese „False Prediction Rate *FPR*“ für die klinische Akzeptanz entscheidend sein, da jeder Fehlalarm mit unnötigen Beeinträchtigungen einhergeht, und bei einer Häufung von falschen Vorhersagen auch die korrekten Alarme vom Patienten ignoriert würden. Daher muss eine maximale Fehlerrate $FPR_{\max}$ eingeführt werden, die als noch tolerabel angesehen wird. Als eine obere Grenze für $FPR_{\max}$ kann die mittlere Anfallshäufigkeit angenommen werden, d. h. es dürfen in einem Zeitraum höchstens so viele Fehlalarme ausgelöst werden, wie auch Anfälle eintreten. Während der prächirurgischen Diagnostik liegt die Anfallshäufigkeit im Mittel bei drei bis vier Anfällen pro Tag, bzw. bei ungefähr 0.15 Anfällen pro Stunde. Dieser Wert hat sich für erste Abschätzungen der Leistungsfähigkeit von Vorhersageverfahren als sinnvoller Anhaltspunkt erwiesen [21].

Zur Überprüfung der Aussagekraft einer Vorhersagemethode sollte sie auf ihre statistische Signifikanz geprüft werden. Als signifikant wird sie dann betrachtet, wenn die Methode sich als besser als eine rein zufällige Vorhersagemethode erweist. Zu diesem Zweck wurde ein „Random Predictor“ definiert [24], bei dem von zufällig ausgelösten Alarmen ausgegangen wird. Viele dieser Alarme werden falsch sein, jedoch werden – zufällig – auch einige Alarme den Anfällen im korrekten Zeitraum vorausgehen, und daher als richtige „Vorhersage“ klassifiziert werden. Dem „Random Predictor“ wird wie auch der zu untersuchenden Vorhersagemethode eine maximale Fehlerrate $FPR_{\max}$ zugestanden, was in Anhang A.3 ausgeführt ist.

Beispielsweise können für Werte von $SOP = 40$ Minuten und $FPR_{\max} = 0.15$ Fehlalarme pro Stunde bereits bis zu drei von zehn Anfällen durch den „Random Predictor“ zufällig vorhergesagt werden. Damit das Ergebnis einer Vorhersagemethode als signifikant eingestuft werden kann, muss sie bei gleichen Parametern für SOP und $FPR_{\max}$ eine höhere Sensitivität als der Random Predictor erzielen.

2.5 Zusammenfassung

Die Vorhersage epileptischer Anfälle könnte eine vielversprechende akute Behandlung für ansonsten kaum therapierbare Patienten ermöglichen. Damit Anfälle unterdrückt werden können, müssen Vorhersagemethoden frühzeitig Alarme auslösen, um anschließend eine zeitlich gezielte Intervention durchzuführen. Als aussichtsreich haben sich Methoden der Nichtlinearen Dynamik erwiesen, mit deren Hilfe charakteristische Zeitreihen aus aufgezeichneten EEG-Daten berechnet werden können.

Entscheidend für die Bewertung von Vorhersagemethoden ist eine systematische, statistische Analyse ihrer Ergebnisse. Es müssen die erzielte Sensitivität und die maximal zugelassene Rate an Fehlalarmen gegeneinander abgewogen werden. Zu diesem Zweck wurde das formalisierte Bewertungskriterium „Seizure Prediction Characteristic“ vorgestellt, anhand dessen die Leistung beliebiger Vorhersagemethoden in Abhängigkeit definierter Vorhersagefenster beurteilt werden kann.

Kapitel 3

Konzeptioneller Entwurf der Software-Umgebung

Für die Anwendung der im vorherigen Kapitel dargestellten Methoden zur Anfallsvorhersage und deren Bewertung wird nun der Entwurf für eine Software-Umgebung ausgearbeitet. Besonders zur systematischen Auswertung von Langzeit-EEG-Daten ist eine automatisierte Anwendung der Verfahren unerlässlich. Dabei stellen sich sowohl von methodischer als auch von technischer Seite spezifische Anforderungen an dieses Software-Projekt. Nach der Bestimmung derselben wird in diesem Kapitel der im Rahmen dieser Diplomarbeit entwickelte Entwurf des Analyse-Prozesses dargestellt. Ferner werden die nötige Berechnung der Kenngrößen aus dem EEG, sowie das Kernprogramm „SeizureAnalyzer“ beschrieben. Näher eingegangen wird außerdem auf die verwendeten Entwurfstechniken, mit Hilfe derer die im Folgenden ausgearbeiteten Anforderungen umgesetzt wurden.

3.1 Zentrale Anforderungen

3.1.1 Flexibler Einsatz und Erweiterbarkeit

Die Software-Umgebung wird zur Erforschung unterschiedlicher Methoden der Vorhersage epileptischer Anfälle eingesetzt. Hierbei werden sich die Einsatzbedingungen der Software praktisch immer unterscheiden. Daher sind übersichtlich strukturierte Konfigurationsmöglichkeiten sowie eine einfache Erweiterbarkeit zentrale Anforderungen. Dies betrifft speziell

- die Einbeziehung zusätzlicher Patienten in die Analyse, deren Daten und Zusatzinformationen zusammen mit weiteren Parametern berücksichtigt werden müssen;

- Parameter, welche die Vorhersage steuern, wie z. B. die minimale und maximale Länge der Vorhersagefenster (siehe 2.4), oder die maximalen Fehlerraten, für welche die Vorhersagesensitivität untersucht werden soll;

- technische Parameter, über welche u. a. die Behandlung von problembehafteten Datenstücken definiert werden kann – z. B. die maximale Länge von Lücken in den Daten, welche für die Vorhersage hingenommen werden soll;

- die Konfiguration der nötigen Vorverarbeitungsschritte, und die Erweiterbarkeit um zusätzliche Vorverarbeitungsmodule;

- die Integration zusätzlicher Vorhersagemethoden.

Sowohl auf Implementationsebene als auch auf Anwendungsebene sind daher verschiedenste Schnittstellen für Anpassungen und Erweiterungen vorzusehen.

3.1.2 Benötigte Typen der Vorhersagemethoden

Die Vorhersagemethoden müssen zuerst für einzelne Patienten trainiert werden, um anschließend angewendet und analysiert werden zu können. Hierbei ist eine Online-Vorhersage interessant, welche einer echten Anwendung nahe kommt. Diese wird in Abschnitt 3.1.4 besprochen.

3.1.2.1 Optimierungsmodus

Die betrachteten Vorhersagemethoden enthalten diverse freie Parameter, für welche optimale Werte bestimmt werden müssen, um gute Vorhersageleistungen zu ermöglichen. Als optimal wird jener Parametersatz betrachtet, über den eine maximale Zahl von korrekten Vorhersagen beobachtet wird – bei einer Fehlerrate von höchstens $FPR_{\max}$. Diese „besten“ Parameter müssen für alle Patienten einzeln ermittelt werden, da sich in den bisherigen Forschungsarbeiten herausgestellt hat, dass es nicht möglich ist, für alle Patienten gleichermaßen sinnvolle Parametersätze zu bestimmen. Auch werden von den gemäß 2.3 berechneten Features nur ein oder zwei beste für

die Vorhersage verwendet, welche im Zuge der Optimierung auszuwählen sind. Die wichtigsten dieser für jeden Patienten zu optimierenden Parameter sind:

- Ein bester **Schwellwert** für jedes betrachtete Feature.
- Dasjenige **Feature**, welches die besten Vorhersagen ermöglicht.
- Der für den Patienten optimale **Interventionszeitraum** *SPH*.

Die Optimierung muss entweder auf den kompletten aufgezeichneten Daten jedes Patienten, oder auf ausgewählten Trainingsdatensätzen durchgeführt werden können. Die Auswahl der optimalen Schwellwerte muss automatisch verlaufen, während die Vorhersageergebnisse der verschiedenen Features einzeln auszugeben sind, so dass die Auswahl eines „besten Features“ auf Wunsch auch manuell erfolgen kann. Im Gegensatz zum „Seizure Prediction Horizon“ wird die „Seizure Occurence Period“ nicht automatisch bestimmt, sondern entsprechend der konkreten Rahmenbedingungen extern festgelegt.

3.1.2.2 Testmodus

Falls optimale Vorhersageparameter für einen Patienten bestimmt wurden, z. B. durch eine vorhergehende Optimierung gemäß dem letzten Abschnitt, kann in einem Testmodus die Überprüfung der Vorhersage mit diesen festen Parametern erfolgen. Hierbei dürfen nicht dieselben Daten sowohl zur Optimierung als auch zum anschließenden Test verwendet werden. Daher wird typischerweise die erste Hälfte der vorliegenden Aufzeichnungen für die Optimierung, und die zweite Hälfte für die Überprüfung der optimierten Methode verwendet. Dies entspricht im Groben der zukünftigen Anwendung: im Rahmen einer „Vorhersage-Diagnostik“ könnten EEG-Daten erhoben werden, anhand derer die besten Parameter für einen Patienten zu bestimmen sind. Im Falle zufriedenstellender Ergebnisse wäre anschließend eine Übertragung der optimierten Vorhersageparameter auf ein Implantat möglich. Um abzuschätzen, ob die Vorhersage dann auch anhand unbekannter Daten hinreichend gute Ergebnisse erzielt, kann sie im Testmodus durchgeführt werden.

3.1.3 Kombination von Vorhersagemethoden

Für eine mögliche Verbesserung der Vorhersagemethoden bietet es sich an, mehrere Methoden zu kombinieren. Diese analysieren parallel die Daten und lösen gemäß

ihrer Methodik Alarme aus. Vom Gesamtsystem wird dann ein Alarm signalisiert, wenn z. B. von allen beteiligten Methoden innerhalb eines vorgegebenen Zeitfensters Alarme ausgelöst wurden (logische *UND*-Verknüpfung). Unter der Annahme, dass die von den Vorhersagemethoden ausgelösten Fehlalarme unabhängig voneinander auftreten, könnte durch eine *UND*-Verknüpfung die Rate der Fehlalarme reduziert werden, während hingegen die Zahl der korrekt vorhergesagten Anfälle jenen der einzelnen Methoden entsprechen sollte.

3.1.4 Online-Vorhersage

Bisher stehen zur Optimierung und Überprüfung der Vorhersagemethoden die Daten zur Verfügung, die im Zuge der prächirurgischen Diagnostik aufgenommen wurden (siehe 2.2). Da diese Aufzeichnungen retrospektiv vorliegen, ist eine Vorhersage in Echtzeit derzeit nur in einem Simulationsmodus möglich. Dennoch ist es wünschenswert, das Vorhersagesystem entsprechend zu gestalten, um in Form einer simulierten Online-Auswertung die praktische Durchführbarkeit von vielversprechenden Methoden zu testen und eventuell demonstrieren zu können. Hierbei dürfen nur Daten und Informationen aus der „Gegenwart“ und Vergangenheit verwendet werden.

3.2 Der Analyse-Prozess

Die Analyse von Methoden zur Vorhersage epileptischer Anfälle mit Hilfe retrospektiv vorliegender EEG-Aufzeichnungen gliedert sich in verschiedene Phasen. So sind als Erstes die Daten aufzubereiten, um aus diesen die charakteristischen Features berechnen zu können, welche anschließend zur Vorhersage verwendet werden. Abschließend müssen die Vorhersageergebnisse aufbereitet und ausgewertet werden.

Es ist wünschenswert, den gesamten Analyseprozess möglichst durchgängig zu automatisieren, um systematische Auswertungen für breite Parameterbereiche leicht durchführen zu können. Die Analyse-Schritte sollten direkt aufeinander aufbauen und über Programmgrenzen hinweg die Informationen der bereits vorher gewonnenen Ergebnisse automatisch berücksichtigen. Dabei müssen die wesentlichen Konfigurationsmöglichkeiten leicht angepaßt werden können.

In Abbildung 3.1 ist ein Überblick über den Analyse-Prozess sowie die beteiligten Informationen bzw. Daten dargestellt. Die Phasen des Prozesses werden im Folgenden besprochen.

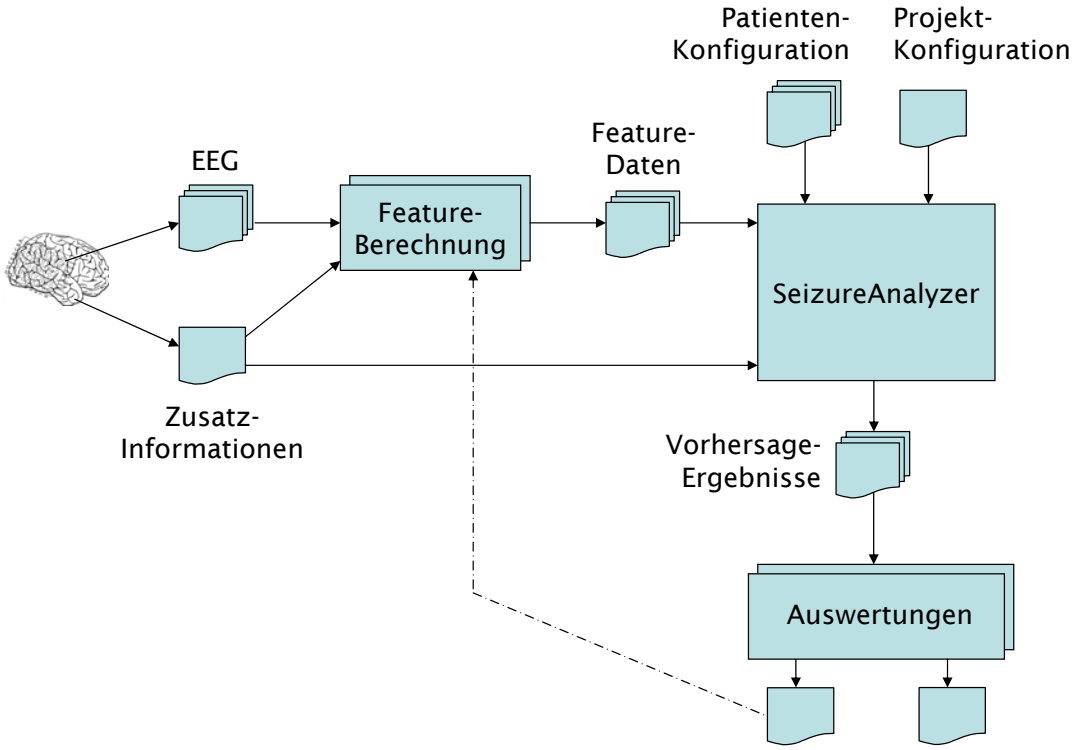


Abbildung 3.1: Überblick über den Analyseprozess: unter Berücksichtigung von zusätzlichen Informationen werden aus den EEG-Daten Features berechnet, die vom Programm „SeizureAnalyzer“ zur Vorhersage-Analyse genutzt werden. Letztere führt das Programm entsprechend der Patienten- und Projekt-Konfiguration durch. Die ermittelten Ergebnisse können weiter ausgewertet werden, um z. B. mögliche Verbesserungen für die Feature-Berechnung zu untersuchen.

Wie in Abschnitt 2.3 dargestellt werden aus den aufgezeichneten EEG-Daten charakteristische Features gewonnen, welche spezifische Veränderungen im EEG messen und zum Zwecke der Anfalls-Vorhersage geeignet scheinen. Von den in der Literatur beschrieben Verfahren [4] werden in dieser Arbeit der „Phasensynchronisationsindex“ und der „Dynamical Similarity Index“ verwendet.

Für die Aufbereitung der Daten sowie für die Berechnung dieser Indices lagen bereits bewährte und zuverlässige Skripte und Programme vor, die mit nur kleinen Modifikationen zur Auswertung der Langzeit-EEG-Daten benutzt werden konnten. Daher war eine Konzentration auf die Entwicklung eines Programmes zur Anwendung und Analyse der Vorhersagemethoden möglich.

Genauso wie die medizinische Behandlung sehr spezifisch auf die einzelnen Patienten abgestimmt werden muss, so müssen auch die Vorhersagemethoden individuell optimiert werden, um möglichst gute Anfallsvorhersagen treffen zu können. Dies betrifft einerseits die Höhe der kritischen Schwellwerte, bei denen Alarme ausgelöst werden, als auch die Auswahl des für die Vorhersage besten Features. Daher ist diese Analyse für einen Patienten als eigenständiger Block der Ausführung zu sehen, der für alle betrachteten Patienten iteriert werden muss.

Neben der Optimierung von Vorhersageverfahren müssen auch weitere Analysen durchgeführt werden können. Für diesen Zweck wurde das Programm „SeizureAnalyzer“ entworfen, welches die flexible Einbindung von verschiedenartigen Methoden ermöglicht. Im nächsten Abschnitt wird der Entwurf des Programms näher dargestellt.

Neben der Beurteilung ihrer Leistungsfähigkeit sollen aus den Ergebnissen der verwendeten Vorhersagemethoden auch Rückschlüsse auf potenzielle Ansätze zur Verbesserung der Verfahren ermöglicht werden. Diesbezüglich müssen die Vorhersage-Resultate entsprechend der interessanten Fragestellungen ausgewertet werden. Es existiert bisher z. B. noch keine systematische Untersuchung, ob auch für lange Interventionszeiträume *SPH* von einigen Stunden sinnvolle Vorhersagen erzielt werden können. Um dies abzuklären, können mit Hilfe des „SeizureAnalyzer“ die Vorhersagemethoden für variable Werte von *SPH* angewandt werden. Anschließend ist zu überprüfen, für welche Interventionszeiträume signifikante Ergebnisse zu beobachten sind.

Der ausgearbeitete Analyseprozess bietet hierzu einen flexiblen Rahmen, innerhalb dessen auf nachvollziehbare Art und Weise die Überprüfung und Verbesserung von Vorhersagemethoden durchgeführt werden kann.

3.3 Entwurf des Analyseprogrammes „SeizureAnalyzer“

Die zentrale Aufgabe des im Rahmen dieser Diplomarbeit entwickelten Software-Projektes liegt in der Optimierung und Anwendung der verschiedenen Vorhersageverfahren. Hierfür wurde das Programm „SeizureAnalyzer“ entsprechend der ermittelten Anforderungen entworfen und implementiert. Der Entwurf wird nachfolgend vorgestellt.

3.3.1 Objektorientierte Programmierung

Um den Zielen der Modularität und Flexibilität gerecht zu werden, bietet es sich an, die Software objektorientiert zu entwerfen. Unter einem Objekt ist hierbei eine zusammengefasste, strukturierte Einheit von spezifischen Daten und zugehörigen Operationen zu verstehen [25]. Die Menge aller gleichartiger Objekte wird als eine so genannte Klasse definiert, in der die Struktur der Daten sowie die zur Verfügung stehenden Methoden festgehalten sind. Die Klassen geben also die generelle Funktionalität der Objekte als Schablone vor, während Objekte spezielle Ausprägungen „ihrer“ Klassen sind.

Eine Möglichkeit zur Erweiterung der Funktionalität einer Klasse besteht in der Vererbung ihrer Eigenschaften an eine Unterklasse, welche um neue Methoden angereichert sein kann. Gleichzeitig kann die Unterklasse als Erbin auf die Methoden der ursprünglichen Klasse zugreifen, so dass diese nur einmal definiert werden müssen. In der Praxis bedeutet dies, dass die allgemeine Funktionalität eines Programmteils als Oberklasse entworfen wird, und über verschiedene erbende Unterklassen spezielle Erweiterungen und Veränderungen vorgenommen werden.

Der Programmablauf erfolgt bei objektorientierten Programmen durch Interaktionen zwischen den beteiligten Objekten, welche jeweils eine spezifische Aufgabe haben. Weitergehende Aufträge werden an andere Klassen delegiert, indem deren Operationen verwendet werden, um „Nachrichten“ auszutauschen.

Sehr sinnvoll kann dabei die Definition einer abstrakten Oberklasse sein, in welcher durch die Angabe einer Menge von Operationen eine „Schnittstelle“ eingeführt wird. Diese Klasse wird daher als „Interface“ bezeichnet. Sie muss erst von abgeleiteten Klassen mit entsprechender Funktionalität realisiert werden, welche dann aber über

„ihr Interface“ angesprochen werden können. Durch diese Kapselung sind externe Klassen von den Details der konkreten Implementierungen unabhängig.

Unter anderem wurde ein Interface für die allgemeine Analyse von Vorhersagemethoden erstellt, welches für jeden Patienten benutzt wird. Spezielle Methoden können dann als realisierende Klassen des Interface verwendet werden, ohne dass den externen Klassen Details bekannt sein müssen, die über jene hinaus gehen, welche durch das Interface spezifiziert sind.

Der Einsatz dieser objektorientierten Entwurfstechniken für das Programm „Seizure-Analyzer“ wird im Folgenden vorgestellt.

3.3.2 Übersicht über die Programmstruktur

Der grundlegende Ablauf des SeizureAnalyzers besteht aus den folgenden Schritten:

- Lesen der Daten eines Patienten.
- Anwendung der gewünschten Vorhersagemethode(n).
- Speicherung der Vorhersageergebnisse.

Die ersten beiden Schritte werden iteriert, so dass die Daten blockweise nacheinander gelesen und ausgewertet werden. Aufgrund der retrospektiven Analyse wäre es auch möglich, die Daten vollständig einzulesen und erst anschließend für die Vorhersage zu verwenden, was aber in Hinblick auf die zukünftige Echtzeit-Analyse unvorteilhaft wäre. Ferner wurden die Blockgrößen nicht festgelegt, so dass sie optimal an die durchzuführende Vorhersagemethode angepaßt werden können.

Umgesetzt wurden die ermittelten Anforderungen gemäß Abbildung 3.2, in welcher die Programmstruktur als *UML*-Klassendiagramm (*Unified Modeling Language* [26]) dargestellt ist. Für grundlegende Aufgaben der Initialisierung und Ablaufkontrolle wurde die Klasse AnalysisFramework vorgesehen. Diese benutzt die Klasse PatientAnalyzer, welche die Analyse der gewünschten Vorhersagemethode auf den Daten eines Patienten durchführt. Dazu greift sie auf die Klasse DataReader zu, um die Daten der gewünschten Datenquellen einzulesen, sowie auf die Interface-Klasse DataProcessor. Über abgeleitete Klassen dieses Interfaces können die verschiedenen Vorverarbeitungs-Algorithmen implementiert werden.

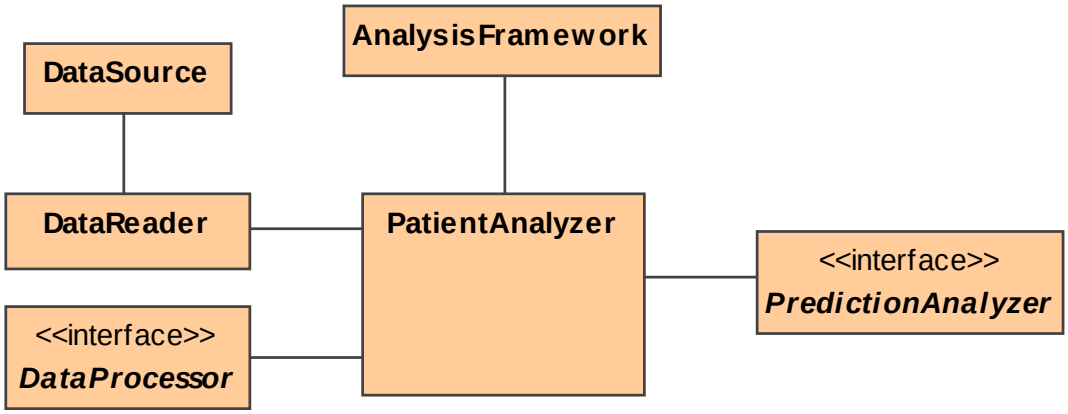


Abbildung 3.2: *UML*-Klassendiagramm der wesentlichen Module des „SeizureAnalyzer“. Zentral ist die Klasse PatientAnalyzer, die vom AnalysisFramework benutzt wird, um die Vorhersage-Analyse für einen Patienten durchzuführen. Dazu werden die Daten über die Klasse DataReader eingelesen, welche über das Interface DataProcessor vorverarbeitet werden können. Dabei können Feature-Datenquellen verwendet werden, die über die Klasse DataSource modelliert wurden. Das Interface PredictionAnalyzer dient der Einbindung der eigentlichen Methode zur Analyse der Vorhersage.

Zur Durchführung der eigentlichen Vorhersage-Analysen wurde die Interface-Klasse PredictionAnalyzer entworfen. Dieses definiert die grundlegenden Operationen, über welche die Vorhersagemethoden als abgeleitete Klassen eingebunden werden können.

Durch die Verwendung der Interface-Klassen wird erreicht, dass z. B. die Klasse PatientAnalyzer von der jeweils eingesetzten Vorhersagemethode unabhängig ist (siehe auch Abschnitt 3.3.1). Jede Klasse, welche eine Analysemethode implementiert und dabei dem Interface PredictionAnalyzer entspricht, kann anhand dieses Interfaces in abstrakter Weise verwendet werden.

In den folgenden Abschnitten wird detailliert auf den Entwurf der Programmkomponenten eingegangen.

3.3.3 Datenverarbeitung

3.3.3.1 Module zum Einlesen der Feature-Daten

Wie im Abschnitt 2.3 erläutert, werden bei den betrachteten Vorhersagemethoden verschiedene charakteristische Features als Indikatoren für bevorstehende Anfälle berechnet und ausgewertet. Deren Anzahl und Art muss flexibel gehalten werden, um neben den derzeit verwendeten auch zusätzliche einbinden zu können, bzw. um die Kombination mehrerer Features zu ermöglichen (siehe 3.1.3).

Die abstrakten Eigenschaften der Datenquellen werden durch die Klasse DataSource gekapselt. Jede Datenquelle enthält dabei einen oder mehrere der Features. Mindestens eine Datenquelle muss angegeben werden, welche dann von der Klasse DataReader ausgelesen wird. Letztere enthält die nötigen Funktionen zum sequentiellen Lesen beliebig großer Datenblöcke. Hierbei muss auf eine sehr robuste Funktionsweise geachtet werden, um einen reibungslosen Betrieb sicherzustellen. Gelegentlich auftretende Probleme in den Daten, wie z. B. fehlende oder fehlerhafte Einträge, müssen automatisch erkannt werden, so dass eine anschließende Fehlerbehandlung durchgeführt werden kann. Ferner müssen auch die Zeiträume angegeben werden können, für welche die Vorhersagemethoden angewandt oder ausgelassen werden sollen. Zum Beispiel erzeugen die als Teil der prächirurgischen Diagnostik durchgeführten Elektrostimulationen Artefakte in den EEG-Daten, welche die Ergebnisse verfälschen. Aus diesem Grund dürfen die Zeiträume der Stimulationen inklusive einer Nachlaufzeit nicht in die Vorhersage-Analyse mit einbezogen werden. Die Implementation des Programmes muss daher möglichst anpassungsfähig sein, ohne jedoch Abstriche bei der Zuverlässigkeit der Datenverarbeitung zuzulassen.

3.3.3.2 Vorverarbeitung

Die berechneten Feature-Daten können nicht direkt für die Vorhersagen genutzt werden, sondern müssen vorverarbeitet werden. So müssen die Daten normalerweise geglättet werden, da sie zuviel Rauschen enthalten.

Um generell verschiedenste Vorverarbeitungsverfahren einbinden zu können, wurde das Interface DataProcessor entworfen. Wie in Abbildung 3.3 dargestellt wird es von der Klasse PatientAnalyzer verwendet, um von DataProcessor abgeleitete Klassen zu benutzen, welche die Algorithmen zur Vorverarbeitung implementieren.

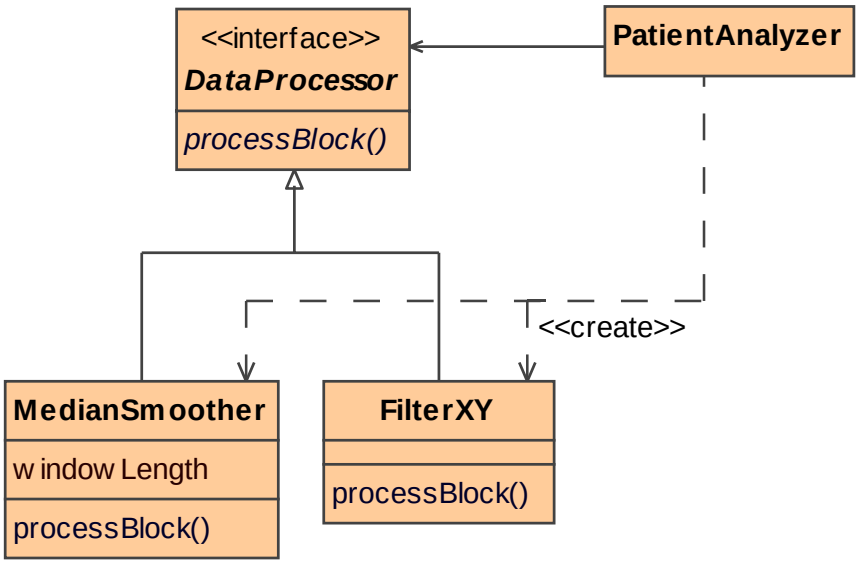


Abbildung 3.3: Klassendiagramm des Interfaces DataProcessor und der realisierenden Klassen MedianSmoother sowie (beispielhaft) FilterXY, welche von PatientAnalyzer erzeugt werden. Über das Attribut windowLength wird ein Parameter des Median-Glätters angegeben, für FilterXY sind beliebige andere Parameter möglich. Die Verwendung der Vorverarbeitungs-Klassen erfolgt über das Interface.

Je nach Anforderung erzeugt die Klasse PatientAnalyzer z. B. ein Objekt der Klasse MedianSmoother, über dessen Operation ProcessDataBlock() die Median-Glättung für einen Datenblock vorgenommen werden kann. Nach der Erzeugung spricht es das Objekt jedoch nur noch über das DataProcessor-Interface an, wie auch weitere benötigte Filter, so dass durch die einheitliche Einbindung der Filteralgorithmen von deren Details abstrahiert wird. Für die Benutzung neuer Filtertypen sind daher keine Änderungen am PatientAnalyzer notwendig.

3.3.4 Durchführung der Analyse für Daten eines Patienten

Zum Starten und Steuern der zu analysierenden Vorhersagemethoden mit den Daten eines einzelnen Patienten ist die Klasse PatientAnalyzer vorgesehen. Von dieser werden spezifische Patienten-Parameter eingelesen, und die Daten des Patienten bestimmt, welche über die Klasse DataReader (siehe 3.3.3.1) gelesen werden können. Hierbei müssen die Daten auf bestimmte Zeitabschnitte beschränkt werden können.

Von der Klasse PatientAnalyzer werden sequentiell die gewünschten Zeitblöcke eingelesen, und anschließend von den konfigurierten Vorverarbeitungsmethoden verarbeitet.

Entsprechend der ausgewählten Art der Vorhersage muss von PatientAnalyzer die Klasse eingebunden werden, welche von PredictionAnalyzer abgeleitet diese Vorhersagemethode implementiert. Die vorverarbeiteten Daten werden über die Funktion AnalyzeNewBlock() der Vorhersagemethode übergeben, die daraufhin die Analyse des Datenblocks beginnt.

3.3.5 Entwurf von Vorhersagemethoden

Als Vorhersagemethoden werden Algorithmen zur Prädiktion epileptischer Anfälle bezeichnet, welche gemäß Abschnitt 2.4 eine Intervention ermöglichen. Diese beinhalten Methoden, um aus den aufgezeichneten Daten charakteristische Kenngrößen zu bestimmen, anhand derer Vorhersagen getroffen werden können. Da das Programm „SeizureAnalyzer“ auf den bereits berechneten Feature-Daten arbeitet, kann von den Verfahren der Zeitreihenanalyse abstrahiert werden. Das Programm muss zur Verarbeitung der Features passend konfiguriert werden, um diese einlesen und auswerten zu können.

Für den Einsatz in Online-Verfahren wurde ein grundlegendes Vorhersagesystem entwickelt, welches als Zustandsmaschine entworfen ist. Dies ermöglicht eine explizite Definition des gewünschten Verhaltens, wie das Vorhersagesystem auf eintretende Signale wie Alarme oder Anfälle zu reagieren hat. Als weiterer Sicherheitsaspekt ist daher kein Übergang in einen undefinierten Zustand möglich. Der Entwurf des Vorhersagesystems wird im folgenden Abschnitt beschrieben.

Zur Anwendung, Optimierung und Überprüfung verschiedener Arten der Vorhersage wurden Klassen entworfen, welche das Interface PredictionAnalyzer realisieren. Durch eine entsprechende Änderung der Konfiguration des Programms kann dann ausgewählt werden, welche Vorhersageart bzw. welche Klasse benutzt werden soll.

3.3.5.1 Entwurf des Vorhersagesystems als Zustandsmaschine

Die Kernkomponente der Online-Vorhersage ist die Klasse PredictionSystem (siehe Abbildung 3.4), welche das Vorhersagesystem als Transitionssystem modelliert. Es wird eingesetzt, um auf Ereignisse und Signale, die bei der Analyse der Daten eintreten, anhand des aktuellen Systemzustandes zu reagieren. Im Gegensatz zur herkömmlichen Art der Verarbeitung kommt dies einer Online-Analyse nahe, bei der aufgrund laufend erhobener Daten in Echtzeit Zustandsübergänge ausgelöst werden. Dabei treten folgende Ereignisse ein:

- **alarm**: Durch die Überschreitung eines Schwellwertes ausgelöster Alarm.
- **seizure**: Beginn eines epileptischen Anfalls des Patienten.
- **Zeitüberschreitung**: Ausgelöst durch Ablauf eines definierten Zeitfensters (im Diagramm mit after() gekennzeichnet)

Als Zustände sind definiert:

- **InterIctal**: Der Ruhezustand zwischen Anfällen.
- **PreIctalSPH**: Zustand nach Auslösung eines Alarmes während des Interventionszeitraums SPH.
- **PreIctalSOP**: Akuter Alarmzustand, während dessen mit einem Anfall gerechnet werden muss.
- **PostIctal**: Zustand nach einem Anfall.

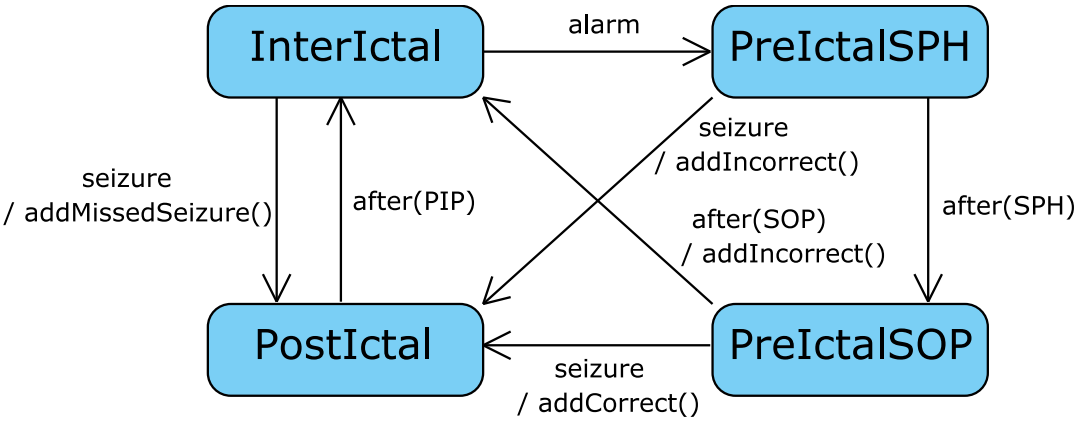


Abbildung 3.4: Zustandsdiagramm des Vorhersagesystems: Durch einen Alarm wird vom Ruhezustand InterIctal ein Übergang zum Zustand PreIctalSPH ausgelöst. Falls nach Ablauf der Zeitdauer *SPH* im Zustand PreIctalSOP ein Anfall eintritt, wird die Vorhersage über die Aktion addCorrect() als korrekt eingeordnet. Wenn während des Zustands PreIctalSPH ein Anfall eintritt, oder im Zustand PreIctalSOP im Verlauf der Zeitdauer *SOP* der vorhergesagt Anfall nicht eintritt, wird der Alarm als falsch eingeordnet. Nach jedem Anfall geht das System in den nach-Anfalls-Zustand PostIctal über, und nach Ablauf der *PostIctal Period* (*PIP*) zurück in den Ruhezustand InterIctal.

Der interiktale Zustand entspricht dem Ruhezustand des Vorhersagesystems. Bei Auslösung eines Alarms durch Schwellwertüberschreitung geht es in den Zustand PreIctalSPH über, in welchem eine Intervention erfolgen soll. Daher muss der Alarm als Fehlalarm klassifiziert werden, falls während PreIctalSPH ein Anfall eintritt. Nach Ablauf des Zeitraumes *SPH* muss entsprechend Abschnitt 2.4 während des nachfolgenden Zeitfensters *SOP* mit einem Anfall gerechnet werden. Falls während dieses Zustandes PreIctalSOP ein Anfall eintritt, so ist der Alarm als korrekt zu klassifizieren, ansonsten handelt es sich hingegen um einen Fehlalarm.

Nach einem Anfall wird das Vorhersagesystem für den Zeitraum *PostIctal Period* in einen Wiederherstellungs-Zustand versetzt, während dessen im EEG noch atypische Aktivität zu beobachten ist. Alarme werden während dieses Zustands nicht ausgelöst. Falls weitere Anfälle im PostIctal-Zustand folgen, verlängert sich der Zeitraum entsprechend.

Durch verschiedenste Ursachen kann es zu Unterbrechungen des Vorhersagesystems kommen. Speziell muss mit Lücken in den Feature-Daten gerechnet werden, die z. B. durch Aufzeichnungspausen entstehen, wenn Patienten vorübergehend von den Aufzeichnungs-Geräten getrennt werden. Dies hat zur Folge, dass jene Anfälle nicht vorhersagbar sind, für deren Alarmzeitraum die Daten nicht vollständig vorliegen. In

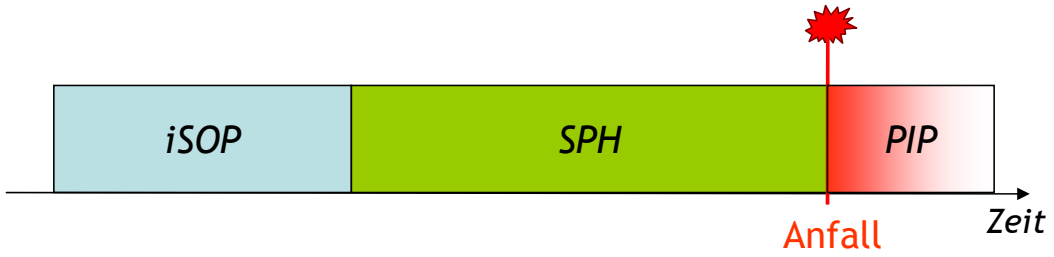


Abbildung 3.5: Darstellung der Vorhersagefenster, von einem Anfall aus betrachtet: inverse *Seizure Occurence Period iSOP* (= Alarmzeitraum), *Seizure Prediction Horizon SPH* und *PostIctal Period PIP*. Während des Zeitraumes *iSOP* muss ein Alarm ausgelöst werden, um den anschließenden Anfall korrekt vorherzusagen.

Abbildung 3.5 ist dieser als inverse *Seizure Occurence Period iSOP* dargestellt, also als Zeitraum, in dem die Alarme liegen können, welche den anschließenden Anfall korrekt vorhersagen würden. Auch jene Anfälle können nicht vorhergesagt werden, welche zu dicht einem anderen Anfall folgen, denn mindestens die Zeit $PIP+SOP+SPH$ muss zwischen ihnen liegen. Um diesen Details der Online-Vorhersage gerecht zu werden, müssen einige Begriffe eingeführt bzw. genauer bestimmt werden:

- **Vorhersagbare Anfälle:** Alle Anfälle des Patienten, die einen Mindestabstand $PIP+SOP+SPH$ zum vorhergehenden Anfall haben, und für die während des Alarmzeitraumes kontinuierliche Feature-Daten vorliegen.

- **Fehlalarm-Referenzzeit:** Summe aller Zeiträume, in denen Fehlalarme auftreten können. Fehlalarme können während der *iSOP*- und *PIP*-Zeiträumen aller Anfälle nicht auftreten, denn Alarme während des *iSOP*-Fensters sind korrekte Vorhersagen, und während der *PIP* werden keine Alarme ausgelöst.

Die verfeinerten Definitionen der Sensitivität sowie der Spezifität (als „False Prediction Rate“ *FPR*) sind hiermit:

- **Sensitivität:** Anteil der sowohl vorhersagbaren als auch vorhergesagten Anfälle an der Gesamtzahl aller vorhersagbaren Anfälle.

- **Spezifität:** Rate der Gesamtzahl aller Fehlalarme pro Fehlalarm-Referenzzeit.

Anhand dieser Bestimmung der relevanten Begriffe ist eine fest umrissene Implementierung des Vorhersagesystems möglich, wie sie für die Realisierung der Online-Vorhersage im nächsten Kapitel durchgeführt wird. Sie erlaubt eine kontinuierliche

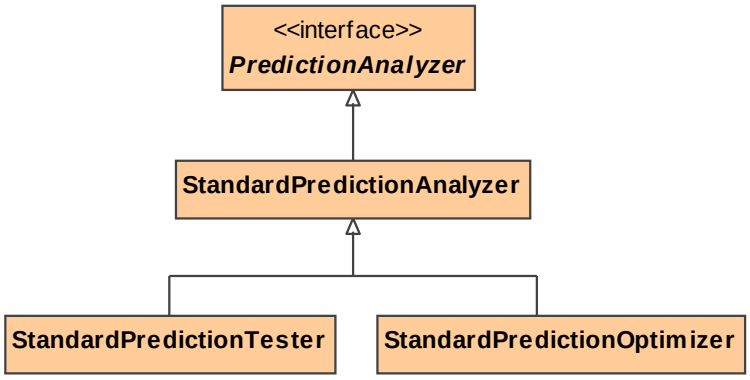


Abbildung 3.6: Klassendiagramm des Interfaces PredictionAnalyzer und exemplarischer, realisierender Klassen zur Analyse, Anwendung und Optimierung einer Vorhersagemethode.

Auswertung der Features, bei welcher direkt Alarme ausgelöst werden. Online wird geprüft, ob diesen im *SOP*-Fenster korrekt ein Anfall folgt. Mithilfe der vorgenommenen Definitionen kann anschließend anhand der beobachteten Sensitivität und Spezifität die Vorhersageleistung beurteilt werden.

3.3.5.2 Modellierung der Schnittstelle zur Einbindung der Vorhersagemethoden

Entsprechend Abbildung 3.2 wurde das Interface PredictionAnalyzer als Schnittstelle für die verschiedenen Vorhersagemethoden entworfen. Es definiert die Methoden StartAnalysis und FinishAnalysis zur Initialisierung und Finalisierung der Vorhersagemethoden, und die Methode AnalyzeNewBlock. Über diese werden von der Klasse PatientAnalyzer die vorverarbeiteten Daten (siehe 3.3.3.2) übergeben.

Abbildung 3.6 zeigt eine mögliche Verwendung des Interfaces PredictionAnalyzer durch abgeleitete Klassen, welche verschiedene Arten der Analyse und Optimierung von Vorhersagemethoden implementieren. Dargestellt ist dies für die Analyse einer Standard-Vorhersagemethode, welche als Klasse StandardPredictionAnalyzer entworfen ist. Über die abgeleiteten Klassen StandardPredictionTester sowie StandardPrediction-Optimizer kann die Vorhersagemethode getestet bzw. optimiert werden (siehe Abschnitt 3.1.2).

Auch hier können durch Erweiterung oder Austausch der Analyse-Klassen auf einfachem und systematischen Wege andere Analyse- oder Vorhersagemethoden ein-

gebunden werden, indem diese so gestaltet werden, dass sie über das Interface PredictionAnalyzer verwendet werden können.

3.4 Zusammenfassung

Sowohl von der Seite möglicher medizinischer Anwendungen als auch aus technischer Sicht ergeben sich vielfältige Anforderungen an eine Software zur Analyse von Vorhersagemethoden. In diesem Kapitel wurde nach der Einführung derartiger Rahmenbedingungen der Entwurf für die Software-Umgebung vorgestellt, welche als flexible und erweiterbare Plattform konzipiert wurde. Zu diesem Zweck wurden objektorientierte Entwurfstechniken eingesetzt. So werden Interface-Klassen verwendet, um abstrahierte Schnittstellen für Erweiterungen zur Verfügung zu stellen.

Eingebettet in einen systematischen Analyse-Prozess ist das Programm „SeizureAnalyzer“ der zentrale Bestandteil der Software-Umgebung, über welche die Analyse der Vorhersagemethoden durchgeführt wird. Hierfür wurde eine Zustandsmaschine entworfen, die das Verhalten des Vorhersagesystems in Bezug auf eintretende Signale in Abhängigkeit vom aktuellen Systemzustand definiert.

Kapitel 4

Implementierung des „SeizureAnalyzer“

4.1 Überblick

Nachdem im vorherigen Kapitel die konzeptionellen Grundlagen für die Entwicklung der Software-Umgebung gelegt wurden, wird nun die Umsetzung in eine lauffähige Anwendung dargestellt. Hierbei erfolgt eine Konzentration auf das Programm „SeizureAnalyzer“, da dieser Teil des Analyse-Prozesses grundlegend neu gestaltet werden musste. Für die vorhergehende Berechnung charakteristischer Zeitreihen aus dem EEG konnte auf bereits bestehende Programme zurückgegriffen werden, und für die Auswertung und Darstellung der ermittelten Ergebnisse des „SeizureAnalyzer“ wurde das Mathematikpaket „Matlab“ verwendet.

Der Programmkern des SeizureAnalyzer wurde dem Entwurf entsprechend realisiert (siehe 3.3.2). Als zentrale Klassen übernehmen AnalysisFramework und AnalysisProject die Initiierung und Verarbeitung der Analyse-Konfiguration. Die Analyse-Parameter werden in der Datenstruktur AnalysisProject abgelegt, wie es in Abbildung 4.1 dargestellt ist.

Für die Verarbeitung der Daten der Patienten und die Durchführung der Analyse ist die Klasse PatientAnalyzer zuständig. Dabei greift diese über Schnittstellen auf Module zum Einlesen bzw. Vorverarbeiten der Daten und zur Anwendung der Vorhersagemethoden zurück, weshalb sie von den Details dieser Module unabhängig ist. So wurden zusammenhängende Klassen als Vorhersage-Module entwickelt, welche über das Interface PredictionAnalyzer zu verwenden sind. Diese werden in Abschnitt 4.4 erarbeitet.

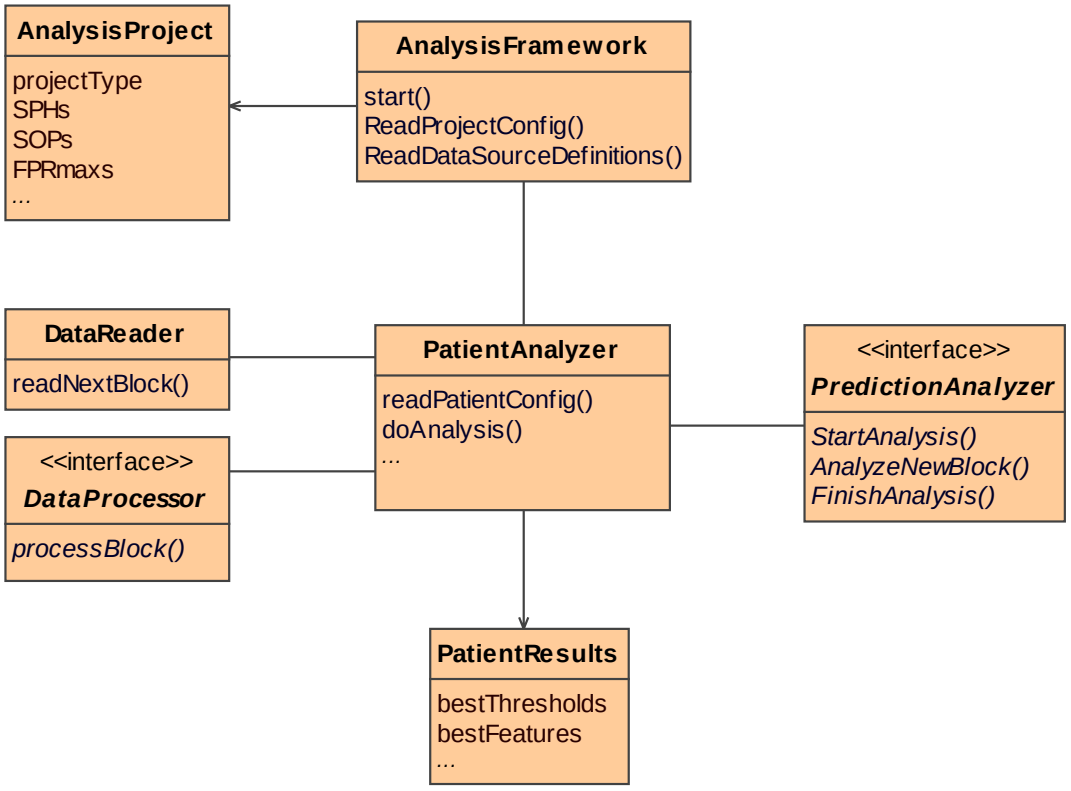


Abbildung 4.1: Diagramm der Kernkomponenten des SeizureAnalyzer, einschließlich deren zentralen Attribute und Funktionen. Die Rahmenklasse AnalysisFramework ist zuständig für die Initialisierung und der Verarbeitung der angegebenen Projektoptionen, welche über die Klasse AnalysisFramework verwaltet werden. Zur Durchführung der Vorhersage liest die Klasse PatientAnalyzer mit Hilfe des DataReader die Feature-Daten blockweise über die Funktion readNextBlock() ein, welche über das dargestellte Interface DataProcessor vorverarbeitet werden können. Die Daten können anschließend über ein Vorhersage-Modul analysiert werden, das über des Interface PredictionAnalyzer eingebunden wird. Die Ergebnisse werden in der Klasse PatientResults zusammengeführt.

Um der Anforderung der flexiblen Einsetzbarkeit gerecht zu werden, wurden diverse Konfigurationsmöglichkeiten vorgesehen. Hierbei ist eine klar definierte und dokumentierte Struktur sehr wichtig, da Feinheiten der Vorhersagemethoden und ihrer Analyse für den Sinn ihrer Verwendung entscheidend sind. Zu diesem Zweck wird die „Extensible Markup Language“ *XML* verwendet, wie im folgenden Abschnitt dargestellt wird.

4.2 Konfiguration

4.2.1 Die „Extensible Markup Language“ *XML*

Die „erweiterbare Beschreibungssprache“ *XML* ist ein Standard für maschinen- und menschenlesbare Dokumente [27]. Als so genannte *Metasprache* erlaubt *XML* die Definition von eigenen Dokument-Formaten, welche aus einzelnen Elementen zu einer Baumstruktur aufgebaut werden. Für Anwendungen, die Dokumente im *XML*-Format verwenden, kann eine spezielle Dokumentenstruktur z. B. durch eine „Document Type Definition“ *DTD* festgelegt werden. Hierdurch ist die Struktur klar formuliert, und der Aufbau von Dokumenten kann gegen diese *DTD* validiert werden, so dass ihre formale Korrektheit überprüft werden kann.

Besonders wichtig sind im Bereich medizinischer Anwendungen Dokumentenformate, die möglichst langfristig von Computern und Menschen interpretiert werden können. Dies kann durch die Definition eigener Formate erreicht werden, welche eine sichere Grundlage für die Datenspeicherung bieten. Aus diesem Grund ist das *XML*-Format auch bei Projekten mit ähnlichen Anforderungen verbreitet (siehe z. B. [28, 29]).

Zur Veranschaulichung ist in Auflistung 4.1 ein beispielhaftes *XML*-Dokument gezeigt. Nach der Deklaration als Dokument der *XML*-Version 1.0 und der Angabe einer Zeichenkodierung folgen die einzelnen Elemente, die durch so genannte „Tags“ (in spitzen Klammern) ausgezeichnet sind. Die Elemente sind baumartig verschachtelt, und können Attribute – wie im Beispiel die Nummer der Patienten – enthalten.

In Auflistung 4.2 ist die entsprechende Definition des Dokumententyps für das Beispiel 4.1 gezeigt. Diese ist wie folgt zu lesen:

- Das Element PatientList enthält einzelne Elemente mit Namen Patient in beliebiger Anzahl (durch den Stern gekennzeichnet).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<PatientList>
  <Patient number="1">
    <Age>59</Age>
    <Gender>f</Gender>
    <NoOfSeizures>34</NoOfSeizures>
  </Patient>

  <Patient number="3">
    <Age>44</Age>
    <Gender>m</Gender>
    <NoOfSeizures>27</NoOfSeizures>
  </Patient>
</PatientList>
```

Auflistung 4.1: Beispiel für ein einfaches *XML*-Dokument, in welchem über hierarchisch strukturierte Elemente exemplarische Informationen zweier Patienten repräsentiert sind.

```
<!ELEMENT PatientList (Patient*)>
<!ELEMENT Patient (Age,Gender,NoOfSeizures)>
    <!ATTLIST Patient number CDATA #REQUIRED>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT Gender (#PCDATA)>
<!ELEMENT NoOfSeizures (#PCDATA)>
```

Auflistung 4.2: Zum *XML*-Dokument 4.1 zugehörige Dokumententypdefinition (*DTD*).

- Jedes Patient-Element enthält Elemente Age, Gender und NoOfSeizures, welche Informationen über den Patienten festhalten, und hat ein Daten-Attribut number.

- Die Elemente Age, Gender und NoOfSeizures enthalten nicht weiter strukturierte Daten (angegeben durch das Schlüsselwort PCDATA).

4.2.2 *XML* als Konfigurations-Format

Speziell auch zum Speichern von Konfigurationsdaten ist die Verwendung von *XML*-Formaten sehr geeignet [30] und inzwischen weit verbreitet. Da verschiedene Parameter, welche die Feinheiten der Vorhersage und ihrer Analyse steuern, auch für die medizinische Relevanz der Methoden ausschlaggebend sind, müssen sie in klar definierten Formaten mittel- bis langfristig abrufbar sein. Zu diesem Zweck wurden für folgende Bereiche Konfigurationsformate definiert:

- Für das **Analyse-Projekt**, zur Spezifikation der allgemeinen Parameter und Optionen, die für eine Analyse verwendet werden.

- Für die Auswahl der **Patienten**, die in die Analyse mit einbezogen werden, und für die Angabe patientenindividueller Zeiträume, deren Daten verwendet werden sollen.

- Zur Angabe der **Daten-Quellen**, für welche die Vorhersage durchgeführt werden soll.

Die Definitionen der Dokumententypen finden sich im Anhang B. Aufgrund ihrer zentralen Rolle wird an dieser Stelle noch näher auf die Projekt-Konfiguration eingegangen, für die ein Beispiel in der Auflistung 4.3 gezeigt ist.

Für das äußerste Element Project wird über das Attribut type die Art der Vorhersagemethode angegeben – im Beispiel handelt es sich um den Optimierungsmodus des Online-Verfahrens. Ferner anzugeben sind die Parameter des maximalen Abstands zwischen zwei Datenblöcken, die noch als kontinuierlich betrachtet werden können. Aufgrund technischer Beschränkungen der Geräte, welche die EEG-Daten aufzeichnen, treten kurze Lücken auf, die aber nicht zu einer Unterbrechung der Vorhersage führen sollen, falls sie kurz genug sind. Der Parameter maximumBlockDistance gibt die maximale Unterbrechungslänge an, für die der Datenstrom als kontinuierlich betrachtet wird, während bei Unterbrechungen kürzer als maximumGap das Vorhersagesystem nicht zurückgesetzt wird.

```
<Project type="OnlineOptimizing" postIctalPeriod="0.5h"
            maximumBlockDistance="10s" maximumGap="10m">

  <ResultsPath>/home/EEGPrevision/analyse/testResults</ResultsPath>
  <InfoPath>/home/EEGPrevision/converted_info</InfoPath>

  <DataSources>
    <DataSource type="psi" name="Phase Synchronisation Index Data">
      <FilesPath>/home/EEGPrevision/phaseSynchro/calcPSI
      </FilesPath>

      <Thresholds direction="negative" stayAboveBelowTime="1s">
        0:0.005:1
      </Thresholds>
    </DataSource>
  </DataSources>

  <DataProcessors>
    <DataProcessor name="MedianSmoother">
      <Parameter name="windowLength">4m</Parameter>
    </DataProcessor>
  </DataProcessors>

  <FPRmaxs>0.1 0.15</FPRmaxs>
  <SOPs>20m:20m:1h</SOPs>
  <SPHs>10m:5m:2h</SPHs>

</Project>
```

Auflistung 4.3: Beispiel für ein *XML*-Dokument zur Projekt-Konfiguration (Details: siehe Text).

Über die Elemente ResultsPath und InfoPath werden die Pfade der Ergebnisse sowie der zusätzlichen Informationen angegeben, und über den Abschnitt DataSources die Datenquellen, welche für die Vorhersage benutzt werden. Neben einem bezeichnenden Namen wird über das Attribut type die Art der Datenquelle ausgewählt, welche in der Datenquellen-Konfiguration (siehe Abschnitt B.3) spezifiziert sind. Zusätzlich zum Pfad, an dem die Daten gespeichert sind (FilesPath), müssen Informationen zu den Schwellwerten (Thresholds) angegeben werden. Mittels des Attributs direction wird festgelegt, ob die Schwellwertdurchstöße von unten nach oben oder umgekehrt erfolgen müssen, und über stayAboveBelowTime die Zeit, über welche die Daten unterhalb bzw. oberhalb des Schwellwertes liegen müssen. Dies erlaubt es, bei nur kurzen Schwellwertüberschreitungen noch keine Alarme auszulösen. Der eigentliche Inhalt des Thresholds-Elementes gibt den Bereich der Schwellen an – im Beispiel liegen sie zwischen 0 und 1.0, mit einer Auflösung von 0.005.

In der Liste der DataProcessors können die Methoden spezifiziert werden, welche zur Vorverarbeitung der Daten verwendet werden sollen. Hier können Module angegeben werden, welche dem entsprechenden Interface (siehe 3.3.3.2) entsprechen, jeweils mit der Angabe der nötigen Parameter. Für den Algorithmus zur Median-Glättung gibt es hier nur den Parameter windowLength, der die Fensterbreite angibt, über welche die Glättung erfolgen soll.

Als Parameter der „Seizure Prediction Characteristic“ sind anschließend die Werte für $FPR_{\max}$, SOP und SPH anzugeben. Neben einzelnen Werten können auch Wertebereiche angegeben werden, für welche die Vorhersagen durchgeführt werden. Über die Schreibweise 10m:5m:1h wird angegeben, dass Werte zwischen 10 und 60 Minuten (mit einem Abstand von 5 Minuten) zu verwenden sind. Zeiten sind prinzipiell als Sekundenwerte anzugeben, wobei auch die Einheiten m und h für die Angabe von Minuten und Stunden zur Verfügung stehen, welche vom Programm umgerechnet werden. Die Fehlerraten werden immer als Rate der Fehlalarme pro Stunde gemessen, und dimensionslos notiert.

Zusammen mit dem im Anhang unter B.1 spezifizierten Dokumententyp ergibt sich sowohl auf syntaktischer als auch semantischer Ebene ein wohldefiniertes Format zur Abspeicherung der Konfigurationen. Da dieses auch für Erweiterungen offen ist, bietet es sich für einen langfristigen Einsatz zur Datenspeicherung an.

4.3 Datenverarbeitung

Wie sich im Zuge der Anforderungsanalyse (Abschnitt 3.3.3) herausgestellt hat, ist die zuverlässige und robuste Verarbeitung der zu analysierenden Daten eines der kritischen Punkte des Projektes. Verschiedenartige Probleme mit den einzulesenden Daten müssen sowohl explizit erkannt, als auch behandelt werden. Soweit es sinnvoll ist, sollte durch Probleme oder Lücken der Daten kein völliger Abbruch der Vorhersagemethoden eintreten, sondern nur eine Unterbrechung erfolgen, während dessen keine Vorhersage möglich ist.

Dem Entwurf entsprechend wurden diese Anforderungen in der Klasse DataReader umgesetzt. Diese wird vom PatientAnalyzer benutzt, welcher den in der Projekt-Konfiguration angegebenen Pfad der angegebenen Datenquelle(n) nach zu verwendenden Dateien durchsucht, und die Namen dieser Dateien dem DataReader übergibt. Dessen Funktion ReadNextBlock() sucht den als nächstes zu verwendenden Datenblock und liest diesen ein – parallel für eventuell mehrere vorliegende Features. Für jeden Patienten einzeln werden als Differenzmenge der explizit eingeschlossenen und ausgeschlossenen Zeiträume jene periodsToUse bestimmt, für welche die Vorhersage erfolgen soll.

Der erste Datenblock beginnt mit dem Startpunkt des ersten Zeitraumes, während die weiteren aufeinander folgen. Im Falle von Unterbrechungen wird der aktuell behandelte Datenblock beendet, und der Beginn des nachfolgenden Zeitraumes ermittelt, ab dem dann ein neuer Block begonnen werden kann.

Die Daten der berechneten charakteristischen Zeitreihen werden standardmäßig spaltenweise in ASCII-Dateien gespeichert. Die meisten der Probleme betreffen Einträge in diesen Dateien, für die entweder keine oder ungültige Daten vorliegen, welche z. B. nicht im angegebenen Wertebereich liegen. In diesem Falle werden die fehlerhaften Daten ignoriert, mit dem nächsten gültigen Datensatz ein nächster Block begonnen und ein Hinweis in die Logdatei geschrieben.

Alle Module, die zur anschließenden Vorverarbeitung der Daten benutzt werden sollen, müssen der Schnittstelle DataProcessor genügen (siehe 3.3.3.2) und insbesondere die Funktion ProcessDataBlock() anbieten, über welche sie ihren Algorithmus auf einen Datenblock anwenden können.

Implementiert, getestet und verwendet wurde ein Median-Filter, der kontinuierlich von einem über die Daten laufenden Fenster die Zentralwerte bestimmt, und sie als Ergebnis-Zeitreihe zurückliefert. Hierdurch werden „Ausreißer“ herausgefiltert, und die Daten entsprechend geglättet. Lücken von weniger als der Hälfte der Fenster-

breite können ausgeglichen werden, indem von vorübergehend weniger Datenpunkten der Zentralwert ermittelt wird. Eine kontinuierliche Auswertung ist äußerst wünschenswert, um die Zahl der Unterbrechungen der Vorhersage so klein wie möglich zu halten.

4.4 Module zur Vorhersage-Analyse

Im Rahmen dieser Diplomarbeit wurden drei grundlegende Arten der Vorhersage implementiert. Diese setzen auf den Feature-Daten auf, die z. B. über den Phasensynchronisationsindex berechnet werden. Die implementierten Vorhersage-Verfahren beinhalten eine konventionelle Offline-Vorhersage, die neu entwickelte Online-Vorhersage sowie ein Verfahren zur Kombination verschiedener Vorhersagemethoden. Gemäß dem Entwurf in Abschnitt 3.3.5 wurden diese objektorientiert als Module mehrerer zusammengehöriger Klassen realisiert. Sie werden im Nachfolgenden vorgestellt.

4.4.1 Offline-Vorhersage

Bei der Offline-Vorhersage werden die Feature-Daten als vollständig vorliegend angenommen, und können daher umfassend verwendet werden. In einem ersten Schritt werden die Daten eingelesen, und die Zeitpunkte der Überschreitungen von Schwellwerten bestimmt, bei welchen Alarme ausgelöst werden.

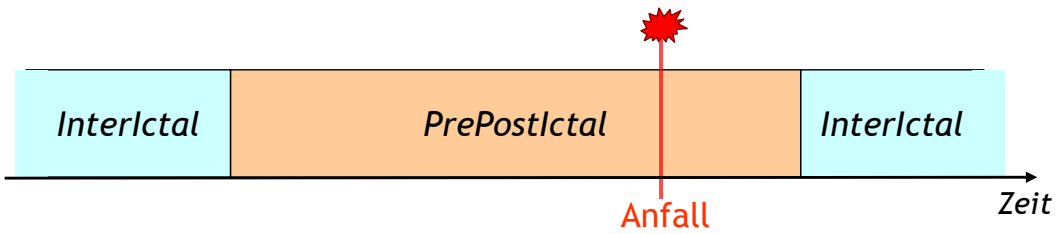


Abbildung 4.2: Einteilung der Aufnahme-Zeit bei der Offline-Vorhersage in interiktale Abschnitte (fern von Anfällen) und jene vor und nach den Anfällen (*PrePostIctal*).

Zur Klassifikation der Alarme wird der Aufnahme-Zeitraum in Abschnitte eingeteilt, in welchen die Alarme als korrekt bzw. falsch eingestuft werden, wie in Abbildung 4.2

gezeigt. Der Zeitraum *PrePostIctal* erstreckt sich über die Dauer des „Seizure Prediction Horizon“ *SPH* und der „Seizure Occurence Period“ *SOP* vor dem Anfall, sowie über die „PostIctal Period“ *PIP* nach dem Anfall. Alarme, die während der interiktalen Periode ausgelöst werden, werden als Fehlalarme klassifiziert.

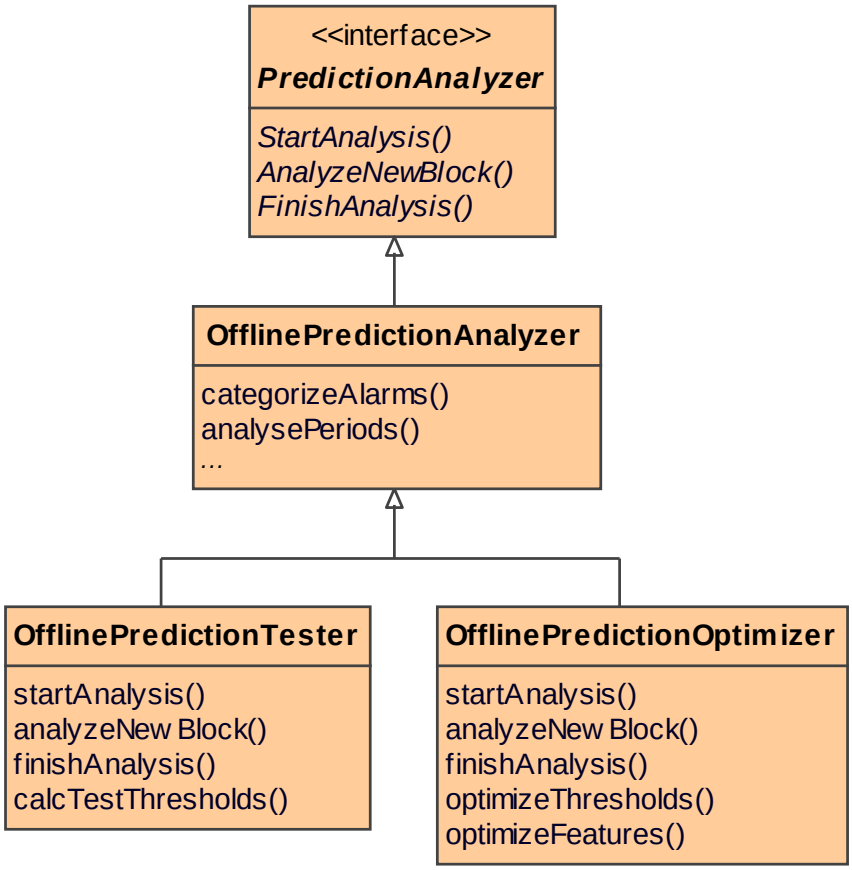


Abbildung 4.3: Klassendiagramm des Moduls zur Offline-Analyse als Realisierung des Interfaces PredictionAnalyzer. Allgemeine Funktionen werden über die Klasse OfflinePredictionAnalyzer zur Verfügung gestellt, deren Erweiterungen auf der einen Seite Funktionen zum Test, aber auf der anderen Seite auch zur Optimierung von Vorhersage-Parametern anbieten.

In Abbildung 4.3 ist die Implementation der Offline-Analyse dargestellt. Die Klassen entsprechen dem Interface PredictionAnalyzer, so dass sie in den SeizureAnalyzer eingebunden werden können. Die grundlegenden Funktionen zur Analyse der Vorhersage-Zeiträume (AnalyzePeriods()) und zur Klassifikation der Alarme (CategorizeAlarms()) werden von der Klasse OfflinePredictionAnalyzer zur Verfügung gestellt. Als Erweiterungen wurden die Klassen OfflinePredictionOptimizer und OfflinePredictionTester abgeleitet,

welche die Methoden zur Optimierung bzw. Überprüfung der Vorhersagen implementieren.

Das grundsätzliche Ziel der Optimierung ist die Bestimmung des Schwellwertes, bei dessen Über- / Unterschreitung maximal viele korrekte Alarme ausgelöst werden – bei Einhaltung der maximalen Fehlerrate $FPR_{\max}$. Die Optimierung muss für die verschiedenen, in der Projekt-Konfiguration vorgegebenen Werte für SOP, SPH und $FPR_{\max}$ einzeln vorgenommen werden, sowie für jede einzelne Zeitreihe. Durchgeführt wird sie über die Funktion OptimizeThresholds() für die ebenfalls in der Projekt-Konfiguration vorgegebenen Schwellwerte, indem die Sensitivitäten und Fehlerraten der Vorhersagen in Abhängigkeit von den Schwellen bestimmt werden. Anschließend wird von den Schwellwerten, mit denen Fehlerraten von höchstens $FPR_{\max}$ beobachtet wurden, jener mit der größten Sensitivität ausgewählt.

Als weiterer Optimierungsschritt wurde die Ermittlung des besten Features implementiert. Durch die Funktion OptimizeFeatures() werden aus den Ergebnissen der optimalen Schwelle jenes Feature ausgewählt, über das die beste Sensitivität erzielt werden kann.

Falls ein Schwellwert fest ausgewählt wurde, kann die Klasse OfflinePredictionTester verwendet werden, um die Vorhersagemethode mit diesem Schwellwert durchzuführen, und die Ergebnisse inklusive Sensitivität und Spezifität berechnen zu lassen. Wie von OfflinePredictionOptimizer werden ebenfalls die Funktionen von der Oberklasse OfflinePredictionAnalyzer zur Kategorisierung der Alarme und zur Analyse der Zeiträume benutzt.

4.4.2 Online-Vorhersage

Für eine Online-Vorhersage sind grundlegenden Änderungen gegenüber dem Modul OfflinePrediction vorzunehmen. Insbesondere müssen die Daten *on the fly* ausgewertet werden, das heißt, sie müssen sukzessive eingelesen und verarbeitet werden. Dabei dürfen nur die Informationen verwendet werden, welche bis zum gerade aktuellen Analysezeitpunkt vorliegen. Ferner muss das Vorhersagesystem instantan auf Signale wie Alarme oder Anfälle reagieren.

Zwar ist ein Online-Betrieb derzeit nur simulationsweise möglich, da bisher eine Auswertung während der laufenden EEG-Aufzeichnung eines Patienten noch nicht durchgeführt werden konnte. Doch soweit die vorliegenden Daten ohne Zugriff auf Informationen aus der „Zukunft“ analysiert werden, kann eine Situation analog zur echten Anwendung erreicht werden. Auch wenn die Implantation eines Vorhersagesystems in

naher Zukunft wohl noch nicht als angemessen erscheint [4], ist es sehr wünschenswert, die bei den diagnostischen Messungen aufgezeichneten Daten von einem externen Computer in Echtzeit für Vorhersagezwecke auszuwerten. Dies ermöglicht erste Untersuchungen der Fragestellung, wie sinnvoll die Übertragung des Vorhersagesystems auf ein Neuroimplantat wäre, von welchem bereits erste Prototypen entwickelt wurden [31]. Letztere können jedoch noch keine Vorhersagen leisten, sondern führen bei Erkennung bestimmter Aktivitätsmuster akute Stimulationen durch.

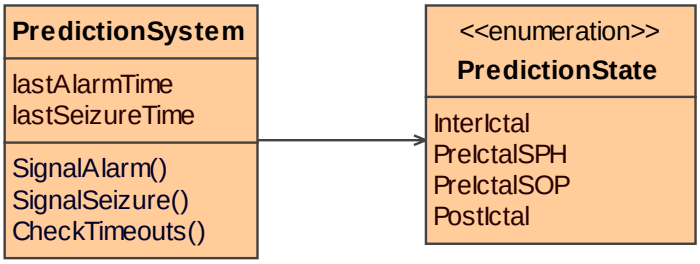


Abbildung 4.4: Klassendiagramm des PredictionSystem, welches das Verhalten der Online-Vorhersage auf eintretende Ereignisse SignalAlarm(), SignalSeizure(), ... definiert. Die möglichen Zustände des Systems sind über die Aufzählung PredictionState angegeben.

Die für die Online-Vorhersage entworfene Zustandsmaschine PredictionSystem wurde durch eine gleichnamige Klasse implementiert, die in Abbildung 4.4 gezeigt ist. Die Zustände sind in der Auflistung PredictionState definiert. Zustandsübergänge werden entsprechend der eingehenden Signale vorgenommen, welche über die Signal-Funktionen bei Eintreten eines Alarms oder Anfalls ausgelöst werden können. Entsprechend des Entwurfes (siehe 3.3.5.1) überprüft die Funktion CheckTimeouts() regelmäßig, ob das Zeitfenster eines Zustandes wie z. B. PostIctalSOP abgelaufen ist, und löst die entsprechenden Zustandsübergänge aus. So wird z. B. nach Ablauf des *SOP*-Fensters der vorherige Alarm als Fehlalarm vermerkt, falls während des Fensters kein Anfall eingetreten ist.

Mittels dieser direkten Abbildung von dem gewünschten Verhalten des Online-Vorhersagesystems auf die Klasse PredictionSystem konnte eine gut nachvollziehbare und überprüfbare Kodierung der Zustandsübergänge erreicht werden, wie sie im Anhang in Auflistung C gezeigt ist. Auch können Veränderungen und Erweiterungen auf einfache und direkte Weise vorgenommen werden. So können bei Bedarf neuartige Zustände integriert werden, und es sind spezielle Varianten der Vorhersage durch abgeleitete Versionen der Klasse möglich.

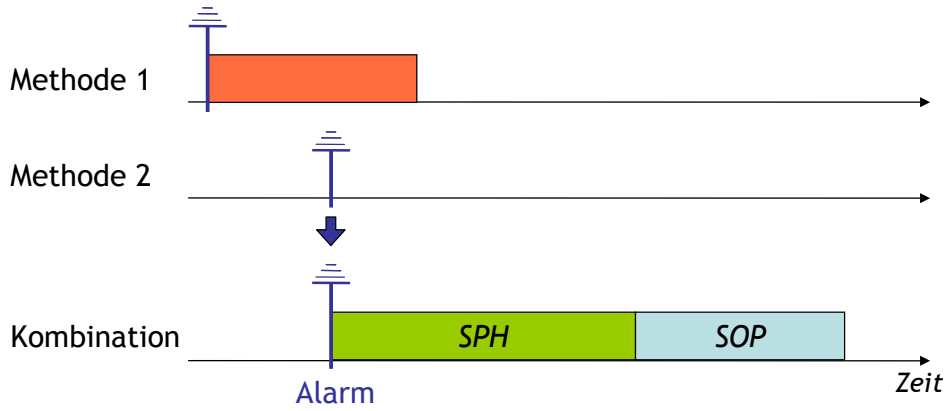


Abbildung 4.5: Kombination von Vorhersagemethoden durch eine logische *UND*-Verknüpfung: falls nach einem ersten Alarm der einen Methode innerhalb eines definierten Zeitfensters auch die andere Methode Alarm schlägt, wird ein Gesamtalarm ausgelöst.

4.4.3 Kombination von Vorhersagemethoden

Für eine mögliche Verbesserung der Vorhersageleistung können verschiedene Methoden kombiniert werden, wie im Abschnitt 3.1.3 dargelegt wurde. Zu diesem Zwecke werden beispielsweise zwei Methoden parallel zur Analyse der vorliegenden Features verwendet, die jeweils bei Schwellwertüberschreitungen einen internen Alarm auslösen.

Dargestellt ist der Ablauf der kombinierten Vorhersage für eine logische *UND*-Verknüpfung zweier Methoden in Abbildung 4.5. Nach der Auslösung eines Alarmes von einer der Methoden beginnt ein Zeitfenster, innerhalb dessen auch die zweite Methode einen Alarm auslösen muss – ansonsten verfällt der erstere. Zum Zeitpunkt dieses zweiten Alarmes wird ein externer Gesamtalarm gegeben, dem die Vorhersagefenster *SPH* und *SOP* folgen. Wie bei der normalen Vorhersage wird der Alarm dann als korrekt klassifiziert, wenn das *SPH*-Fenster anfallsfrei blieb, aber innerhalb des *SOP*-Fensters ein Anfall eintrat.

Auch eine *ODER*-Kombination der Methoden ist vorstellbar, so dass bei jedem einzeln ausgelösten Alarm ein Gesamtalarm ausgelöst wird. Hierbei muss das System so optimiert werden, dass die einzelnen Methoden mit sehr niedriger Fehlerrate arbeiten, da jeder interne Fehlalarm auch zu einem externen Fehlalarm führt. Andererseits reicht zur Vorhersage der Anfälle auch ein einzelner korrekter Alarm, so dass sich die Methoden diesbezüglich ergänzen.

Im Zuge der kombinierten Vorhersage steigt die Anzahl der zu optimierenden Parameter stark an. Falls die verwendeten Methoden z. B. je zehn Features verwenden, sind bereits 50 Kombinationen zweier Features möglich, für die jeweils die Vorhersageparameter angewandt und optimiert werden müssen. Als Heuristik wurde im Rahmen dieser Diplomarbeit daher der Weg gewählt, zuerst in Abhängigkeit von SPH, SOP und $FPR_{\max}$ die beiden besten Features der einzelnen Methoden zu bestimmen. Anschließend können die beiden Schwellwerte so optimiert werden, dass die kombinierte Vorhersage die optimale Sensitivität erzielt, wie immer unter Einhaltung der maximalen Fehlerrate $FPR_{\max}$.

Beim vorhergehenden Schritt zur Bestimmung von besten Features der einzelnen Methoden ist ein Zugriff auf die kompletten Feature-Daten nötig, wodurch dieser Teil des Verfahrens nicht online durchgeführt werden kann. Für die anschließende Anwendung der kombinierten Vorhersage ist dies jedoch möglich, wozu die Klasse PredictionSystem eingesetzt wird.

Implementiert wurde die Kombination von Vorhersagemethoden als abgeleitete Klasse des im vorigen Kapitel besprochenen OnlinePredictionAnalyzer. Entsprechend Abbildung 4.6 ist der CombinedParameterOptimizer um die Funktion DetermineBestFeature() erweitert, über welche die besten Features der einzelnen Vorhersagemethoden ermittelt werden können, sowie um die Funktion OptimizeCombinedThresholds(), welche die für das Gesamtsystem optimalen Schwellwerte der Methoden bestimmen. Die Funktion ProcessSignals() leitet dabei die eintretenden Signale an das PredictionSystem weiter. Im Falle der *UND*-Kombination wird z. B. eben dann einen Alarm ausgegeben, wenn die beteiligten Methoden innerhalb des Zeitfensters interne Alarme ausgelöst haben.

Durch die Erweiterung des OnlinePredictionAnalyzer konnte also auf kanonischem Wege die Klasse CombinedPredictionOptimizer implementiert werden, über welche die Kombination von Vorhersagemethoden optimiert und angewendet werden kann. Diese ist als logische *UND*-Verknüpfung realisiert, wobei darauf aufbauend auch eine Implementierung von weiteren Verknüpfungsformen leicht möglich ist.

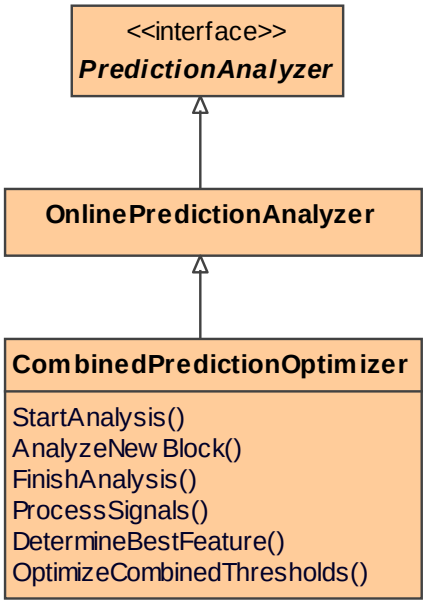


Abbildung 4.6: Diagramm der Klasse CombinedPredictionOptimizer zur Optimierung kombinierter Vorhersagemethoden, abgeleitet von der Klasse OnlinePredictionOptimizer und dem Interface PredictionAnalyzer. Erweitert wurde sie u. a. um die Funktionen OptimizeCombinedThresholds() und DetermineBestFeature().

4.5 Bereitstellung der Ergebnisse für weitere Auswertungen

Für eine einheitliche Abspeicherung der ermittelten Ergebnisse der Vorhersage von einem Patienten wurde die Klasse PredictionResults entworfen, welche in Abbildung 4.1 auf Seite 36 gezeigt ist. In dieser Klasse werden die Ergebnisse abgelegt, welche für die verwendeten Vorhersageparameter erzielt wurden. Dies schließt die Ergebnisse der Vorhersage-Optimierungen mit ein, wie z. B. das ermittelte optimale Feature.

Zur Abspeicherung der Daten wurde neben einem nicht explizit strukturierten *ASCII*-Format auch ein *XML*-Format definiert. In diesem werden jeweils für einen Parametersatz von SPH, SOP und $FPR_{\max}$ die für einen Patienten beobachteten Ergebnisse der Vorhersage gespeichert. Ein Beispiel ist unter Auflistung 4.4 gezeigt, die Definition des Dokumententyps im Anhang B.4. Im Beispiel wurde für Werte von $SOP = 30$ Minuten bzw. 1800 Sekunden und $FPR_{\max} = 0.15$ Fehlalarme pro Stunde die Vorhersage für zwei verschiedene Interventionszeiträume $SPH = 15$ Minuten und 30 Minuten durchgeführt. Die Ergebnisse für einen Parametersatz sind jeweils über das Element PredictionResult zusammengefasst. Die Elemente UsedSeizures und InterIctalTime geben die Anzahl der verwendeten Anfälle an, sowie die Summe der Länge aller Zeiträume zwischen den Anfällen. Das Element Feature bestimmt eventuell optimierte Features, für das bei dem angegebenen Schwellwert (Threshold) eine gewisse Anzahl korrekter und inkorrekter Alarme ausgelöst wurden. Zusätzlich werden die erzielte Fehlerrate FPR und die Sensitivität ausgegeben, optional ferner die genauen Zeitpunkte der Alarme. Zeiten werden dabei immer als Sekundenwerte gespeichert, die Fehlerrate als Anzahl der Fehlalarme pro Stunde interiktaler Referenzzeit (siehe Abschnitt 3.3.5.1).

Durch die Verwendung dieses definierten Formates zur eindeutigen Speicherung der Ergebnisse können diese leicht durch weitere Programme interpretiert werden, welche zur Aufbereitung der Vorhersage-Resultate verwendet werden. Und auch langfristig ist eine Auswertbarkeit der Daten von technischer Seite her gewährleistet.

Um eine einfache und direkte Auswertung der Ergebnisse zu ermöglichen, wurden unter Benutzung des *XML*-Parsers „XML4MAT“ [32] Funktionen zum Import der *XML*-Daten für das Mathematikpaket „Matlab“ entwickelt. Darauf aufbauend konnten durch die Programmierung entsprechender Matlab-Funktionen verschiedene Methoden zur komfortablen Darstellung der erzielten Vorhersageresultate zur Verfügung gestellt werden. Diese erlauben eine automatisierte Aufbereitung der Ergebnisse.

```
<PredictionResults>
    <PredictionResult SPH="900" SOP="1800" FPRmax="0.15">
      <UsedSeizures>23</UsedSeizures>
      <InterIctalTime>436667</InterIctalTime>
      <Feature>12</Feature>
      <Threshold>0.505</Threshold>
      <Corrects>9</Corrects>
      <Incorrects>16</Incorrects>
      <FPR>0.13190</FPR>
      <Sensitivity>0.39130</Sensitivity>
    </PredictionResult>

    <PredictionResult SPH="1800" SOP="1800" FPRmax="0.15">
      <UsedSeizures>21</UsedSeizures>
      <InterIctalTime>432667</InterIctalTime>
      <Feature>34</Feature>
      <Threshold>0.620</Threshold>
      <Corrects>13</Corrects>
      <Incorrects>17</Incorrects>
      <FPR>0.14015</FPR>
      <Sensitivity>0.61904</Sensitivity>
    </PredictionResult>
</PredictionResults>
```

Auflistung 4.4: Beispiel für Vorhersage-Ergebnisse im *XML*-Format. Aufgelistet werden über das Element PredictionResult für jeweils einen Parametersatz (*SPH*, *SOP* und $FPR_{\max}$) die Ergebnisse der durchgeführten Vorhersage (siehe Text).

4.6 Zusammenfassung

In diesem Kapitel wurde die im Rahmen dieser Diplomarbeit entwickelte Realisation des Entwurfes vom Programm „SeizureAnalyzer“ vorgestellt. Das Programm wurde unter weitmöglichster Beachtung der ermittelten Anforderungen angelegt. So wurden für die Verwaltung der Vorhersage-Analysen und für deren Ergebnisse *XML*-Dokumentformate erstellt, deren wohldefinierte Struktur eine langfristige und automatische Auswertung der durchgeführten Analysen ermöglicht. Gleichzeitig sind die Formate für Erweiterungen offen, wodurch Adaptionen an zukünftige Veränderungen im Bereich der Vorhersagemethoden flexibel durchgeführt werden können.

Die Module zum Einlesen und Vorverarbeiten der Feature-Daten wurden ebenfalls mit Hilfe der eingesetzten *XML*-Formate anpassungsfähig gestaltet. Hierbei war wichtig, dass auftretende Unterbrechungen in den aufgezeichneten Daten robust behandelt werden, um die Vorhersage möglichst unterbrechungsfrei durchführen zu können.

Im Zuge des objektorientierten Entwurfs des Programmes konnten verschiedene Verfahren zur Anwendung und Analyse von Vorhersagemethoden als auswechselbare Module implementiert werden. In Hinblick auf zukünftige Vorhersagesysteme, welche in Echtzeit angewendet werden sollen, wurde für die Vorhersage ein Online-Modus entwickelt, bei welchem – einer echten Anwendung entsprechend – die Daten laufend ausgewertet werden. Außerdem wurde ein Modul implementiert, über welches verschiedene Vorhersagemethoden kombiniert werden können, um bessere Vorhersageleistungen zu erzielen. Technische Details der Implementierung sind in Anhang D aufgeführt. Eine Auswertung erster Ergebnisse dieser Vorhersagen erfolgt abschließend im folgenden Kapitel.

Kapitel 5

Auswertungen und Ergebnisse

Das im Rahmen dieser Diplomarbeit implementierte Programm „SeizureAnalyzer“ ermöglicht die Anwendung und Analyse verschiedenster Methoden zur Vorhersage von epileptischen Anfällen. Dies erfolgt auf der Basis von charakteristischen Zeitreihen, die gemäß Abschnitt 2.3 aus Langzeit-EEG-Messungen gewonnen wurden. Zur Darstellung der Leistungsfähigkeit und zur Erläuterung der Arbeitsweise der im vorherigen Kapitel entwickelten Implementation wurden Untersuchungen durchgeführt, deren Ergebnisse in diesem Kapitel präsentiert werden. Die Vorhersagen erfolgten dabei anhand der im nächsten Abschnitt beschriebenen Daten.

5.1 Beschreibung der ausgewerteten EEG-Daten

Im Rahmen einer Kooperation mit dem Neurozentrum der Universitätsklinik Freiburg standen für die Auswertung der Vorhersagemethoden aufgezeichnete EEG-Daten von 14 Patienten zur Verfügung. Wie in Tabelle 5.1 aufgelistet, erstrecken sich die Aufzeichnungen der Patienten über dreieinhalb bis vierzehn Tage, während derer zwischen 13 und 40 Anfälle eingetreten waren. Bei 11 der Patienten wurden während der Messungen Elektrostimulationen vorgenommen, um anhand der Reaktionen der Patienten Rückschlüsse auf die Funktionen der verschiedenen Hirngebiete ziehen zu können. Auch kann auf diesem Wege abgeschätzt werden, ob die Entfernung des Nervengewebes zu kognitiven Beeinträchtigungen führen würde. Die Elektrostimulationen führen dabei zu starken Artefakten im EEG, weshalb die Daten der entsprechenden Zeiträume nicht zur Analyse der Vorhersage verwendet wurden.

Die Daten wurden mit dem digitalen „Neurofile NT“ Video-EEG-System (it-med, Usingen, Deutschland) aufgezeichnet. Es wurden Samplingraten von 256, 512 und

Nr.	Alter	Geschlecht	Dauer [Tage]	# Anfälle	Elektroden	Anfallsart	Lokalisation	Outcome
1	28	m	9.5	28	g, s	SP, CP	NC	Ib
2	41	w	8.5	30	g, s	CP, GTC	NC	IVb
3	50	m	10.4	32	g, s, t	SP, CP, GTC	H	Ib
4	31	w	7.5	16	s, t	SP, CP, GTC	NC	Ia
5	21	m	13.8	13	g, s	SP, CP, GTC	NC	Ia
6	18	m	6.6	28	g, s	SP, CP, GTC	NC	Ia
7	33	w	6.6	15	g, s	SP, CP, GTC	NC	keine Op.
8	45	w	6.4	21	g, s	SP, CP, GTC	NC	IIb
9	34	m	11.8	23	s, t	SP, CP, GTC	H	IIa
10	11	m	7.8	40	s, t	SP, CP, GTC	H	Ia
11	13	m	4.8	21	s	SP, CP	NC	Ib
12	42	w	6.6	18	g	SP	NC	Ic
13	21	w	3.5	18	s, t	SP, CP, GTC	NC	Ia
14	53	w	5.5	18	g, s	SP, CP, GTC	NC	unklassifiziert
∅	31.5	n	7.8	22.9				
Σ			109.5	321				

Abbildung 5.1: Charakteristika der 14 Patienten im Alter zwischen 11 und 53 Jahren, und der EEG-Daten, die von ihnen während der prächirurgischen Diagnostik erhoben wurden. Im Schnitt erstreckten sich die Aufzeichnungen über 7.8 Tage, und es traten 22.9 Anfälle auf. Verwendete Elektroden: Gitter (g), Streifen (s) und Tiefen (t). Anfallsart: einfach-partiell (SP), generalisiert tonisch-klonisch (GTC) und komplex-partiell (CP). Lokalisation: hippokampal (H) oder neokortikal (NC). Outcome: Ergebnis der Operation nach Engel [33]. Bei einem Patienten wurde die Operation nicht durchgeführt, bei einem weiteren wurde das Ergebnis noch nicht klassifiziert.

1024 Hz verwendet, und die Signale mit einem 16 bit Analog-Digital-Wandler digitalisiert. Zur Entfernung längerfristiger Trends wurden Frequenzen kleiner als 0.53 Hz durch einen Hochpassfilter eliminiert, sowie durch einen Bandpassfilter das 50 Hz-Rauschen der Stromleitung.

Von den bis zu 128 vorliegenden EEG-Kanälen wurden von einem Epileptologen jeweils drei Elektrodenkanäle mit Lage innerhalb und außerhalb des epileptischen Fokus ausgewählt, und anschließend gegen einen Kanal mit geringer epileptischer Aktivität referenziert. Entsprechend der Erläuterung in Anhang A.2 wurde für jeden Kanal der Similarity Index berechnet, so dass sich hieraus sechs Features ergaben, und 15 Features aus der Berechnung des Phasensynchronisationsindexes von jeweils zwei verschiedenen Kanälen (siehe Anhang A.1). Diese Langzeit-Features beider Indices stellen eine vergleichsweise große Datenbasis zur Auswertung einzelner oder auch kombinierter Vorhersagemethoden dar.

5.2 Visualisierung der erzeugten Alarme

Ein erster Schritt zur Auswertung von Vorhersageergebnissen besteht in der Darstellung der korrekten und falschen Alarme, um diese visuell inspizieren zu können. Für diesen Zweck wurde im Rahmen dieser Diplomarbeit ein Paket von Funktionen entwickelt, über welches neben den Informationen über die Alarme auf Wunsch auch weitere Details der Vorhersage, sowie auch die verwendeten Daten dargestellt werden können. Ein Beispiel sind die Markerinformationen, welche die Zeitpunkte klinisch interessanter Ereignisse während der diagnostischen Abklärung angeben (siehe 2.2.2), und die interaktiv angezeigt werden können. Dies soll es ermöglichen, neben einer Überprüfung der Korrektheit der durchgeführten Vorhersagen auch Hinweise für Ansätze zur Verbesserung der Methoden zu geben.

Exemplarisch zeigt Abbildung 5.2 für einen Patienten Details für die Vorhersage über das Online-Vorhersagesystem, die für den Zeitraum von mehr als sechs Tagen mit den Parametern $SOP = 30\,\text{min}$, $FPR_{\max} = 0.15$ Fehlalarme/h und $SPH = 12\,\text{min}$ erfolgte. Sie wurde in diesem Fall über die 15 Features des Phasensynchronisationsindexes durchgeführt, von denen die Kombination aus den Kanälen 1 und 6 als beste bestimmt wurde. Gemäß Abschnitt 3.3.5.1 wurden von den 14 eingetretenen Anfällen 6 als „vorhersagbar“ eingestuft, d. h. ihnen gingen keine Anfälle direkt vorher, und es lagen Daten vor. Anhand der ebenfalls automatisch optimierten, eingezeichneten Schwelle bei 0.55 konnten vier der sechs Anfälle korrekt vorhergesagt werden, einhergehend mit einer Fehlerrate von 0.063 Fehlalarmen pro Stunde. Eingetragen wurden auch

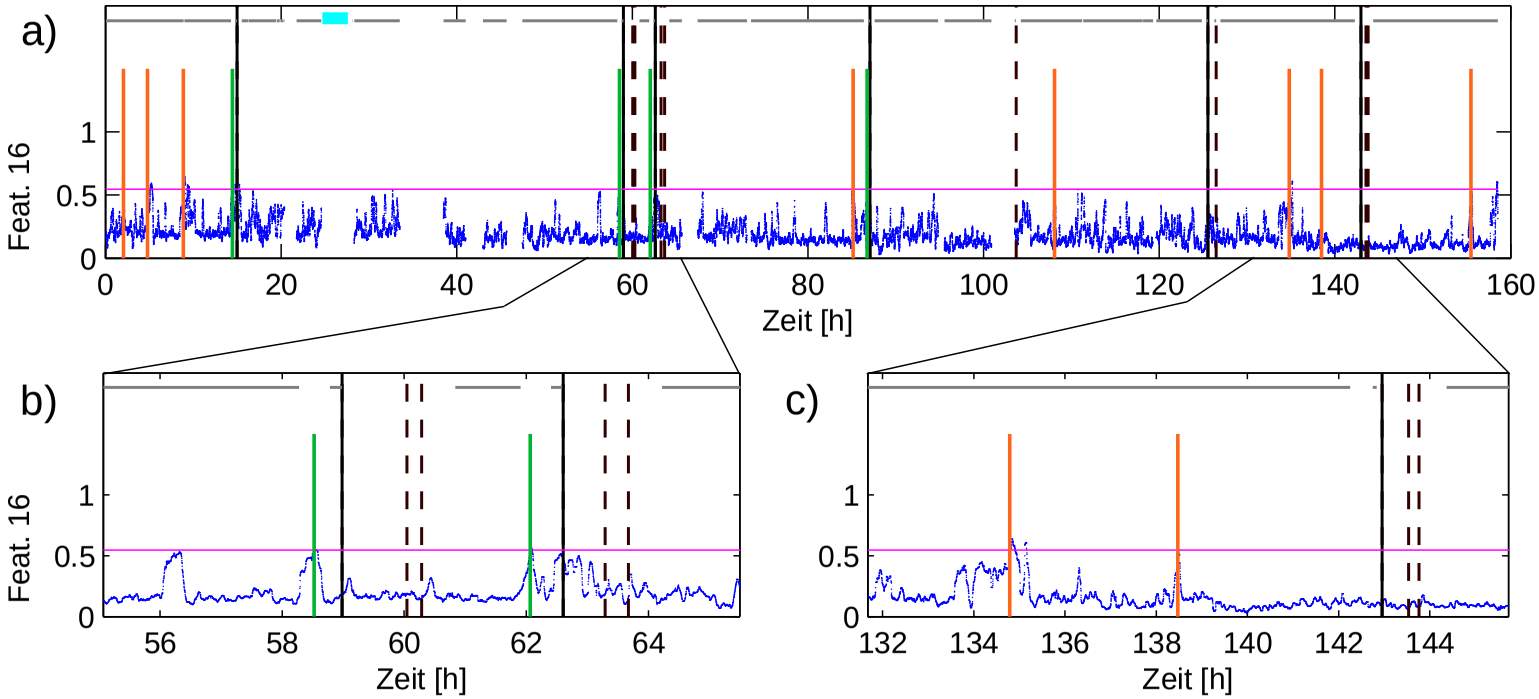


Abbildung 5.2: Beispiel für das Ergebnis der Online-Vorhersage für Patienten Nr. 7, mit bester Schwelle (bei 0.55, purpurrot eingezeichnet) für den Phasensynchronisationsindex zwischen Kanälen 1 und 6 (in blau), SOP=30 min, $FPR_{\max}$=0.15 Fehlalarme/h, SPH=12 min, erreichte Sensitivität=66.7% (vier korrekte Vorhersagen für sechs Anfälle), Fehlerrate=0.063 Fehlalarme/h (acht Fehlalarme in 127.2 h). Die Zeit ist in Stunden seit Anfallsbeginn aufgetragen. Lange, vertikale schwarze Linien markieren die Zeitpunkte der aufgetretenen Anfällen. Jene Anfälle, die als nicht vorhersagbar eingeordnet wurden (siehe Text), sind mit gestrichelter Linie eingezeichnet. Bei Schwellwertüberschreitungen wurden vom Vorhersagesystem korrekte (grün) oder falsche (rot) Alarme ausgelöst. Die oberen, vertikalen grauen Linien kennzeichnen die Zeiträume, in denen Fehlalarme ausgelöst werden können, die türkise Linie den Zeitraum, in welchem elektrisch stimuliert wurde. Die Abbildungen b) und c) zeigen die gekennzeichneten Ausschnitte des vollständigen Zeitraums a).

die ermittelten Zeiträume, in denen Fehlalarme ausgelöst werden können, sowie der Zeitraum der durchgeführten Elektrostimulationen.

Die beiden vergrößerten Abschnitte b) und c) von Abbildung 5.2 stellen Beispiele für korrekte und falsche Vorhersagen dar. Während in b) vor den beiden vorhersagbaren Anfällen Anstiege im Feature zu Schwellwertüberschreitungen und Alarmen führten, folgten den in Abbildung c) gezeigten Schwellwertüberschreitungen innerhalb der passenden Zeitfenster keine Anfällen, so dass diese als Fehlalarme klassifiziert wurden. Der Anfall ca. 143 Stunden nach Aufnahmebeginn konnte nicht vorhergesagt werden, da keine Veränderung im Phasensynchronisationsindex der beiden Kanäle zu beobachten war. Im interaktiven Modus ist es möglich, z. B. für die Zeiträume der ausgelösten Fehlalarme die Markerinformationen anzeigen zu lassen, um auf diesem Wege überprüfen zu können, ob die Veränderungen des Features mit klinischen Ereignissen verbunden waren.

Verschiedene Unterbrechungen der EEG-Aufzeichnung führen zusammen mit den nicht verwendeten Zeiträumen zu Lücken in den Daten des Features, für welche die Vorhersage ebenfalls unterbrochen werden musste – wie in Abbildung 5.2 sichtbar. Da aus diesem Grunde für den im Beispiel ca. 103 Stunden nach Aufnahmebeginn eingetretenen Anfall keine vorhergehenden Daten vorlagen, musste dieser als nicht vorhersagbar eingestuft werden.

Während die visuelle Inspektion der Vorhersagen ein geeignetes Mittel zur Kontrolle der ausgelösten Alarme ist, müssen systematische Untersuchungen für mehrere Patienten durchgeführt werden, um allgemeine Aussagen über die Vorhersageleistungen und ihre statistische Signifikanz treffen zu können. Einige dieser Untersuchungen werden in den folgenden Abschnitten vorgestellt.

5.3 Abhängigkeit der Sensitivität von den Vorhersage-Zeitfenstern

Wie im Abschnitt 2.4 besprochen, bestimmen neben der maximalen Fehlerrate $FPR_{\max}$ insbesondere die Längen der Vorhersage-Fenster SPH und SOP die mögliche Vorhersageleistung. So können durch längere SOP-Fenster mehr Anfälle korrekt vorhergesagt werden, da eben der Zeitraum größer ist, in dem vorhergehende Alarme als korrekt klassifiziert werden. Dies geht jedoch mit einer Abnahme der zeitlichen Präzision der Vorhersage einher, da mit dem SOP-Fenster der Zeitabschnitt größer ist, für den der Anfall vorhergesagt wird - so dass im Endeffekt der Patient für eine längere Zeit in Alarmzustand versetzt ist. Daher wurden Untersuchungen zur Er-

mittlung der Sensitivitäten in Abhängigkeit dieser Parameter durchgeführt, wobei in diesem Beispiel der Wert für $FPR_{\max}$ auf 0.15 Fehlalarme pro Tag festgehalten wurde.

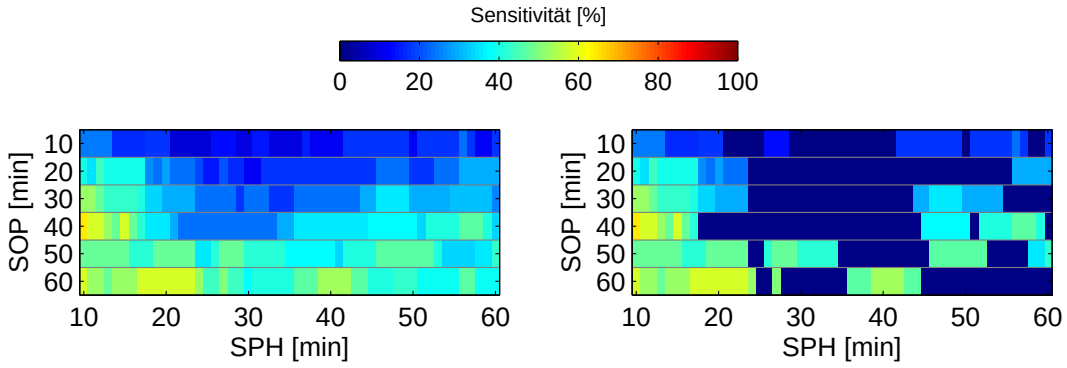
Abbildung 5.3 zeigt beispielhaft für drei Patienten die beobachteten Sensitivitäten für das beste Feature des Phasensynchronisationsindexes, d. h. dessen beste Elektrodenkombination, in Abhängigkeit von den Fensterlängen für *SPH* und *SOP*. Die Vorhersagen wurden über die im vorherigen Kapitel ausgearbeitete Online-Vorhersage des Programms „SeizureAnalyzer“ durchgeführt, und mittels der Klasse OnlinePredictionOptimizer jeweils die optimalen Schwellwerte und Feature bestimmt. Für steigende Fensterlängen von *SOP* zeigen sich erwartungsgemäß steigende Sensitivitäten. Da jedoch auch für den „Random Predictor“ die Anzahl der potentiell korrekten Vorhersagen zunimmt, werden die höheren Sensitivitätswerte teilweise insignifikant, wie auf den Abbildungen in der rechten Spalte zu sehen ist. Es sollten daher die *SOP*-Fenster nicht zu groß gewählt werden, so dass die Patienten nach Möglichkeit genau wissen, wann sie mit einem Anfall zu rechnen haben.

Für variierende Längen des Interventionszeitraums *SPH* konnte kein Wert beobachtet werden, der sich für alle Patienten als optimal erwiesen hat. Für jeden Patienten sollte aus diesem Grund der beste *SPH*-Wert einzeln bestimmt werden, um eine individuell bestmögliche Vorhersage erzielen zu können.

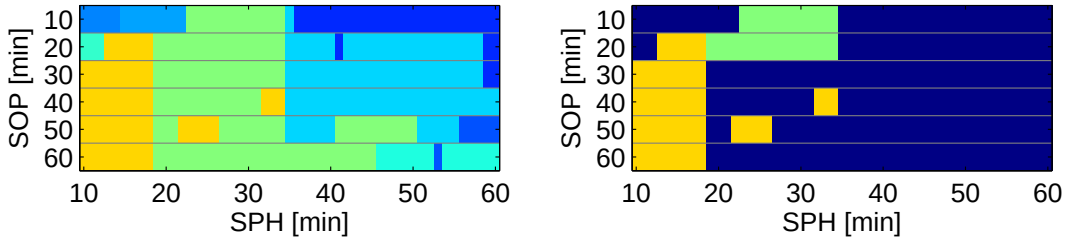
5.4 Kombination zweier Vorhersagemethoden

Zum Zweck der Verbesserung bisheriger Vorhersagemethoden wurde das in Abschnitt 4.4.3 vorgestellte Modul des „SeizureAnalyzer“ implementiert, über welches mehrere Methoden kombiniert werden können. Dieses Verfahren wurde für die vorliegenden Langzeit-EEG-Daten angewendet, wobei als Basis-Vorhersagemethoden der Phasensynchronisationsindex und der Dynamical Similarity Index eingesetzt wurden. Der erste Schritt besteht dabei in der Bestimmung der besten Features für beide Indices. Anschließend werden die Daten jener zwei Features verwendet, und für diese jeweils optimale Schwellwerte bestimmt, so dass bei der kombinierten Auslösung von Alarmen eine maximale Anzahl von Anfällen korrekt vorhergesagt werden. Dies geschieht unter Einhaltung der maximalen Fehlerrate $FPR_{\max}$ durch die Klasse CombinedPredictionOptimizer, welche in Abschnitt 4.4.3 vorgestellt wurde.

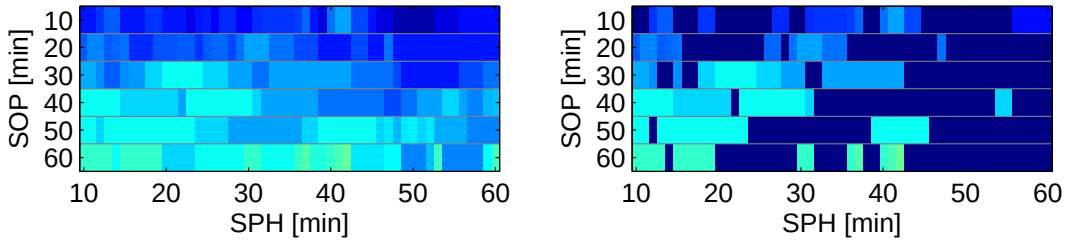
Bei der *UND*-Verknüpfung wird dabei nur dann ein Alarm ausgelöst, wenn innerhalb eines bestimmten Zeitfensters alle beteiligten Vorhersagemethoden Alarm schlagen. Hierfür wurde dieselbe Zeitdauer wie für die „Seizure Occurence Period“ *SOP* gewählt,



a) Patient 1



b) Patient 7



c) Patient 10

Abbildung 5.3: Abhängigkeit der Vorhersage-Sensitivität (in Prozent, farblich kodiert) von den Zeitfenstern *SPH* und *SOP*, exemplarisch für drei Patienten. Dargestellt sind für *SPH*-Werte und *SOP*-Werte zwischen 10 und 60 Minuten jeweils die beobachteten Sensitivitäten für das beste Feature des Phasensynchronisationsindexes. Während in der linken Spalte alle Ergebnisse gezeigt sind, wurden rechts die insignifikanten Sensitivitäten auf 0 gesetzt. Festzustellen sind für längere *SOP*-Fenster steigende Sensitivitäten, welche teils jedoch auch von dem „Random Predictor“ erreicht werden und in diesem Falle nicht signifikant sind. Für variierende *SPH*-Fenster kann kein generell als optimal anzusehender *SPH*-Wert beobachtet werden.

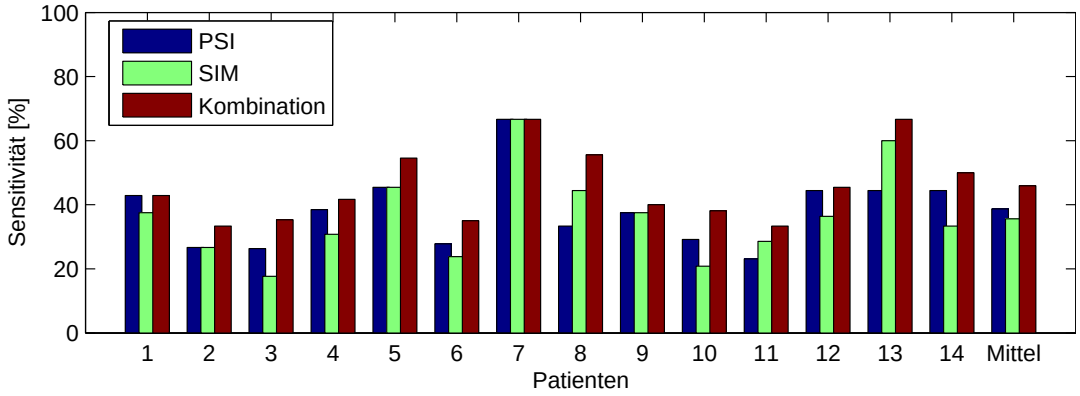
um nach Auslösung eines ersten Alarmes keine längere „Wartezeit“ berücksichtigen zu müssen, als den Zeitraum, für welchen später der Anfall erwartet wird. Getestet wurde dies für eine Zeitdauer von 30 Minuten, und eine maximale Fehlerrate von 0.15 Fehlalarmen pro Stunde. Dargestellt sind die Resultate in Abbildung 5.4a), in der sie mit den einzelnen Ergebnissen des Phasensynchronisationsindex (*PSI*) und des Dynamical Similarity Index (*SIM*) verglichen sind. Bei etwa der Hälfte der Patienten konnte durch die Kombination bessere Sensitivitäten als mit den einzelnen Methoden erreicht werden, und für keinen Patienten wurden schlechtere Ergebnisse als bei diesen beobachtet.

Besonders wichtig ist aber auch in dieser Untersuchung der Test auf die statistische Signifikanz der Ergebnisse, bei der die Vorhersageleistung mit jener eines Prädiktors verglichen wird, der zufällig Alarme auslöst (siehe auch Abschnitt 2.4, [24]). Dabei muss berücksichtigt werden, dass im Zuge der Optimierungen der Vorhersagemethoden vielfach multipel getestet wird. So wurden z. B. aus sechs verschiedenen Werten für den Interventionszeitraum *SPH* von 10 min, 20 min, ..., 1 h für jeden Patienten der optimale Wert ausgewählt. Ferner wurde bei der Bestimmung eines besten Features des Phasensynchronisationsindexes aus 15 Features ausgewählt, beim Similarity Index aus sechs. Für die Kombination wurden von beiden Indices die jeweils besten Features verwendet, so dass insgesamt aus 21 Features ausgewählt wurde.

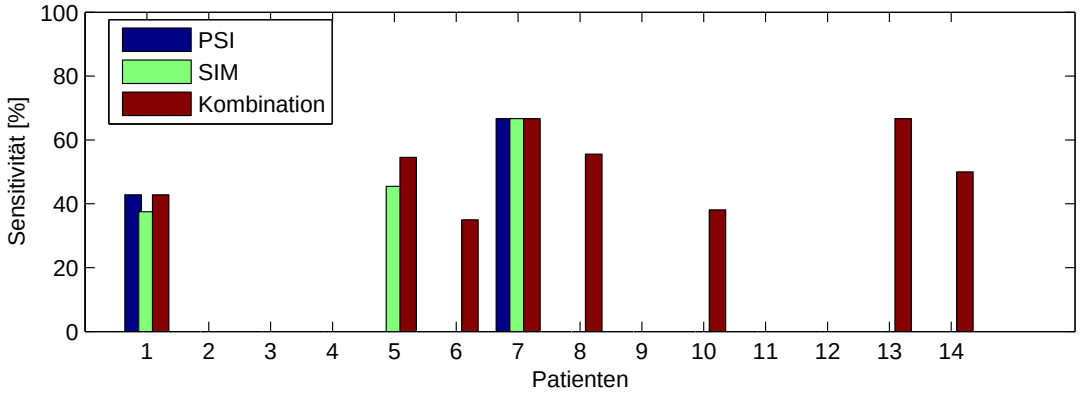
Bei der Berechnung der Sensitivitäten, die vom „Random Predictor“ erreicht werden können, wurde dies entsprechend Anhang A.3 berücksichtigt. Dabei ergaben sich, wie in Abbildung 5.4b) gezeigt, bei den einzelnen Methoden nur für die Patienten 1, 5 und 7 signifikante Ergebnisse. Durch die kombinierte Vorhersagemethoden waren für fünf weitere Patienten Sensitivitäten zu beobachten, die über dem Sensitivitätsniveau des „Random Predictor“ lagen. Während die einzelnen Methoden keine zufriedenstellende Vorhersagen erlauben, scheint die kombinierte Vorhersage ein vielversprechender neuer Ansatz zu sein.

5.5 Lern- und Test-Ergebnisse

Das Ziel der Vorhersage epileptischer Anfälle ist ein Vorhersagesystem, das individuell für einen Patienten optimiert wird, um im anschließenden Betrieb aufgrund fortlaufend gemessener EEG-Daten Vorhersagen zu treffen. Hierbei müssen anhand der optimierten Parameter gute Vorhersageleistungen auch auf den im Folgenden erhobenen Daten erzielt werden. Dies ist im Rahmen der aktuellen Forschung lediglich in beschränkter Weise möglich, da im Vergleich zum langfristigen Betrieb nur relativ kurze, bereits gemessene EEG-Daten ausgewertet werden können. Anhand



a) Ermittelte Sensitivitätswerte.



b) Darstellung der signifikanten Sensitivitätswerte.

Abbildung 5.4: Ergebnisse der kombinierten Vorhersage, durchgeführt über eine *UND*-Verknüpfung des Phasensynchronisationsindex (*PSI*) und des Similarity Index (*SIM*), für Werte von $SOP = 30\,\text{min}$, $FPR_{\max} = 0.15$ Fehlalarme/h und optimierte Werte von *SPH* aus dem Bereich zwischen 10 min und 1 h. Abbildung a) zeigt die beobachteten Sensitivitäten für 14 Patienten, sowie den Mittelwert über alle Patienten, welcher für die Kombination um sieben bis zehn Prozentpunkte über den Sensitivitäten der einzelnen Methoden liegt. Jedoch sind diese nur für einzelne Patienten signifikant, wie in Abbildung b) dargestellt ist. Für fünf Patienten konnten nur mittels der kombinierten Methode signifikante Vorhersageergebnisse beobachtet werden.

dieser Aufzeichnungen können umfassende Überprüfungen der optimierten Vorhersagemethoden noch kaum durchgeführt werden, denn neben der Daten, welche für die Optimierung verwendet werden und möglichst viele Anfälle enthalten müssen, werden weitere lange Abschnitte zur Überprüfung benötigt.

Um jedoch erste Schritte zu einer derartigen Validierung der Trainingsergebnisse zu ermöglichen, wurde als Teil des in Abschnitt 4.4.2 besprochenen Moduls zur Online-Vorhersage eine Klasse OnlinePredictionTester entwickelt, welche die Vorhersage aufgrund vorher optimierter Parameter realisiert. Zur Aufteilung der Daten wurden diese in kurze Abschnitte über einige Stunden geteilt, und zufällig für das Training oder den Test zur Verfügung gestellt. Dies erfolgte unter den Randbedingungen, dass beide eine ähnliche Gesamtzeitdauer der Daten, und eine ungefähr gleiche Anzahl an Anfällen enthielten. Um das Training und den Test mehrmals unabhängig voneinander durchzuführen, wurde die zufällige Aufteilung der Daten wiederholt vorgenommen. Dabei wurden die für jeden Patienten ermittelten Zeiträume über die Patienten-Konfigurationsdatei des „SeizureAnalyzer“ festgehalten. Das Programm wurde anschließend auf den Trainingsdaten zur Optimierung der Vorhersage-Parametern verwendet, um diese in einem zweiten Schritt anhand der Testdaten zu überprüfen.

Exemplarisch sind in Abbildung 5.5 die Ergebnisse für fünf Trainings- und Testläufe eines Patienten farbig kodiert gezeigt. Trotz der erwähnten Beschränkungen konnten bei diesem Patienten für zwei der fünf Läufe auf den Testdaten ähnliche Ergebnisse wie auf den Trainingsdaten beobachtet werden, während für zwei weitere Läufe die Ergebnisse auf eine Sensitivität von 0 einbrachen, und für einen Lauf zwar noch einige Anfälle vorhergesagt werden konnten, doch die Fehlerrate sich auf fast 0.3 Fehlalarme pro Stunde verdoppelte.

Abschließend kann gesagt werden, dass in Zukunft vermehrt derartige Untersuchungen der optimierten Vorhersageparameter auf separaten Testdaten durchgeführt werden müssen, um zu überprüfen, ob die erzielten Ergebnisse auch auf unbekannten Daten erreicht werden können. Während dies bisher nur in wenigen Fällen möglich ist, wurden als Teil dieser Diplomarbeit die technischen Vorraussetzungen für zukünftige Kontrollen geschaffen.

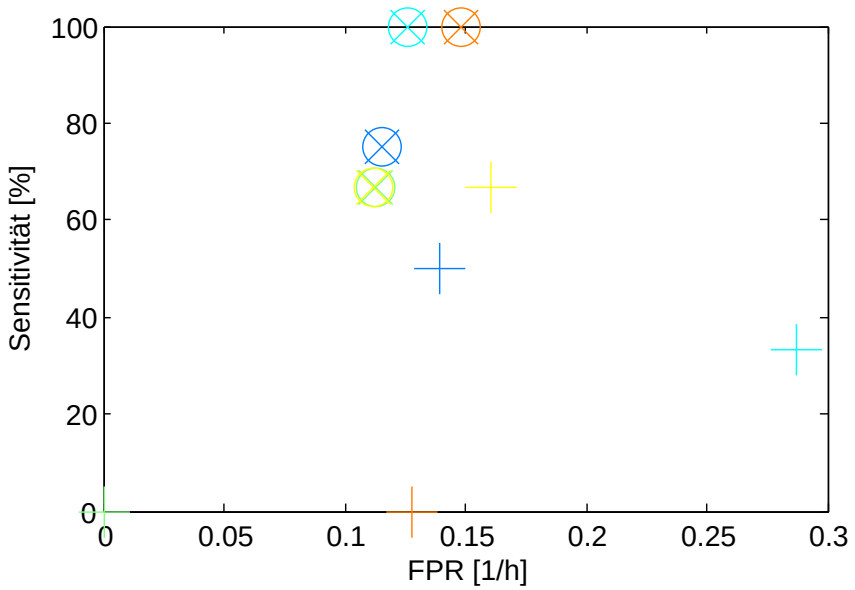


Abbildung 5.5: Beispiel für die Ergebnisse der Vorhersage mit optimierten Parametern (durch ⊗ gekennzeichnet) und deren Anwendung auf Testdaten (+) für Patienten Nr. 7. Eingezeichnet sind für fünf verschiedene, farblich kodiert zusammenhängende Optimierungs- und Testläufe deren Fehlerraten *FPR*, und die erzielten Sensitivitäten. Für die Optimierung wurden als maximal zulässige Fehlerrate 0.15 Fehlalarme pro Stunde erlaubt, *SOP* wurde auf 30 min gesetzt, und für jeden Patienten wurde ein optimaler Wert für *SPH* zwischen 10 min und 1 h bestimmt. Für zwei der Testergebnisse konnten keine Anfälle korrekt vorhergesagt werden (Sensitivität 0, in grün und orange), für eines fiel die Sensitivität auf knapp 35%, bei verdoppelter Fehlerrate (in türkis), und für zwei Läufe (gelb und blau eingezeichnet) konnte im Test eine ähnliche Vorhersageleistung wie auf den Trainingsdaten beobachtet werden.

5.6 Zusammenfassung

Für einige exemplarische Untersuchungen wurde in diesem Kapitel das in dieser Diplomarbeit entwickelte Programm „SeizureAnalyzer“ auf echten Langzeit-Daten angewandt. Bei letzteren handelt es sich um EEG-Messungen von 14 Patienten, die sich über mehrere Tage erstrecken, während derer eine ausreichende Anzahl von Anfällen eingetreten sind, um Methoden zur Vorhersage dieser Anfälle anwenden zu können. Zur Darstellung der aus dem EEG berechneten Features, sowie der ausgelösten korrekten und falschen Alarme wurde ein Paket von Funktionen entwickelt, welches auch eine interaktive, visuelle Überprüfung der erfolgten Vorhersagen erlaubt. Dieses wurde in Abschnitt 5.2 vorgestellt.

Die unter 5.3 gezeigte Ermittlung der Abhängigkeit des Online-Vorhersagesystems von den Längen der Vorhersagefenster *SPH* und *SOP* ergab, dass für die Patienten individuell optimale Interventionszeitdauern *SPH* gefunden werden können, über welche signifikante Vorhersagen möglich sind. Obwohl die im Abschnitt 5.4 durchgeführte Untersuchung für die einzelnen Vorhersagemethoden kaum überzeugende Ergebnisse ergab, konnte durch die Kombination der Methoden eine signifikante Verbesserung beobachtet werden.

Zur Kontrolle, ob die für einen Patienten als optimal bestimmten Vorhersageparameter auch auf Testdaten zu vernünftigen Vorhersageleistungen führen, wurde mit dem entsprechenden Modul des „SeizureAnalyzer“ eine erste Untersuchung durchgeführt, die in Abschnitt 5.5 vorgestellt wurde. Hierbei zeigte sich ein deutlicher zukünftiger Forschungsbedarf, um dieser, für echte Anwendungen kritischen Frage noch weiter nachzugehen, und entsprechende Verbesserungen der Vorhersagemethoden zu ermöglichen. Im Rahmen dieser Arbeit wurden die softwaretechnischen Mittel zur Verfügung gestellt, mittels derer die experimentelle Verwendung und Analyse der Vorhersageverfahren auf realen Langzeit-EEG-Daten durchgeführt werden können.

Kapitel 6

Zusammenfassung und Ausblick

Durch die Möglichkeit der Vorhersage epileptischer Anfälle können verschiedene neue Behandlungsmethoden zur akuten Unterdrückung der überraschend einsetzenden Anfälle anwendbar werden. Automatisierte Verfahren zur Vorhersage von Anfällen sowie zur anschließenden medizinischen Intervention sind insbesondere für Patienten sehr vielversprechend, denen über herkömmliche Therapieansätze nicht oder nur unbefriedigend geholfen werden kann. Zur Erreichung dieses Ziels wurden zwei Methoden vorgestellt, welche charakteristische Informationen aus gemessenen Hirnströmen ermitteln, anhand derer Vorhersagen getroffen werden. Um die Leistungsfähigkeit möglicher zukünftiger Vorhersage-Implantate abzuschätzen, sind systematische Analysen dieser Vorhersagemethoden unerlässlich.

Zu diesem Zweck wurde im Rahmen dieser Diplomarbeit eine Software-Umgebung entwickelt, welche die Durchführung von verschiedenen Vorhersageverfahren auf der Basis von vorliegenden EEG-Daten ermöglicht. Bei letzteren handelt es sich um Langzeit-Messungen der Hirnströme von Epilepsiepatienten, bei denen während mehrtägiger Aufzeichnungen eine ausreichende Anzahl an Anfällen eingetreten sind, damit die automatische Vorhersage dieser Anfälle trainiert und ausgewertet werden kann. Die wichtigsten Grundlagen der Vorhersage epileptischer Anfälle wurden in Kapitel 2 zusammengefasst.

Dabei kristallisierten sich mehrere zentrale Anforderungen an die Software heraus. Neben einer robusten und anpassungsfähigen Verarbeitung der zu verwendenden Daten ist eine ebenfalls flexible Schnittstelle zur Einbindung von Vorhersagemethoden entscheidend, um die Analysen auch für neuartige Varianten durchführen zu können. Mit Hilfe der Verwendung von objektorientierten Entwurfstechniken konnte dies erreicht werden, wie in Kapitel 3 vorgestellt wurde. So wurde ein Modul zur Online-Vorhersage entwickelt, welches aufgrund laufend eingelesener Daten eine Vorhersage in Echtzeit ausführt, entsprechend einer zukünftigen, realen Anwendung.

Die Implementierung des Entwurfes wurde in Kapitel 4 ausgearbeitet. Um eine langfristige Verwendbarkeit der Vorhersage-Analysen und ihrer Auswertungen zu gewährleisten wurde die erweitere Beschreibungssprache *XML* zur Definition zweckmäßiger Dokument- und Datenformate verwendet. Neben der herkömmlichen Vorhersagevariante wurde ein Online-System implementiert, welches eine laufende Auswertung der Daten für die Vorhersage realisiert, sowie auch ein kombiniertes Vorhersageverfahren. Aufgrund der anpassungsfähigen Softwarestruktur können neben diesen auch weitere Vorhersagemethoden über die entworfenen Schnittstellen in das Hauptprogramm eingebunden werden.

Zur Untersuchung der Vorhersage auf den vorliegenden, realen Daten wurden verschiedene Experimente durchgeführt, welche in Kapitel 5 präsentiert wurden. Hierbei wurde die entwickelte Software eingesetzt, um die Online-Vorhersage und die Kombination von Vorhersagemethoden auf Langzeit-EEG-Messungen anzuwenden. Im Vergleich mit einer zufälligen Vorhersagemethode konnten nur teilweise signifikante Ergebnisse beobachtet werden, doch verspricht insbesondere die Kombination deutliche Verbesserungen. Zur Kontrolle der Ergebnisse ist neben der visuellen Inspektion ausgelöster Vorhersagen auch die Frage als wichtig zu erachten, inwieweit die für bestimmte Abschnitte der Daten eines Patienten optimierten Vorhersageparameter auch auf unbekannten Daten sinnvolle Vorhersageleistungen ermöglichen. Mittels der im Rahmen dieser Diplomarbeit implementierten Software-Umgebung können derartige Untersuchungen nun komfortabel und auch gemäß speziellen Anforderungen durchgeführt werden.

Als ein nächster Schritt in Richtung einer klinischen Anwendung der Vorhersagemethoden wäre als Pilotstudie eine Auswertung von EEG-Daten in Echtzeit sehr interessant. Dabei würden die Daten in einem ersten Zeitraum zum Training der Vorhersage genutzt, um anschließend mithilfe der individuell optimierten Parameter eine Online-Vorhersage zu leisten. Auch für diesen Einsatzzweck ist die Software-Umgebung bereits weithin geeignet, da das Vorhersagesystem für einen kontinuierlichen Einsatz entworfen wurde. Ansätze für adaptive Anpassungen der Vorhersageparameter im laufenden Betrieb sind als sinnvolle Erweiterung in Betracht zu ziehen, um in Echtzeit auf Veränderungen der Vorhersagesituation reagieren zu können. Ebenfalls sollten weitere Möglichkeiten zur Kombination von Vorhersagemethoden erforscht werden, da diese deutliche Verbesserungen der Vorhersageleistungen versprechen.

Anhang A

Mathematische Herleitung der verwendeten Vorhersagemethoden

A.1 Phasensynchronisationsindex

Wie in 2.3.1 dargestellt, wird mittels des Phasensynchronisationsindex die Synchronisation zwischen den Phasen zweier schwingender Signale gemessen. Dazu muss aus einem reellen Signal die Phase extrahiert werden. Ein möglicher Ansatz zur Bestimmung der Phase eines reellen Signales $s(t)$ folgt der Definition des komplexen, so genannten *Analytischen Signales*

$$\begin{aligned} s_a(t) &= s(t) + i\hat{s}(t) && \text{(A.1)} \\ &= A(t)e^{i\phi(t)}\,, && \text{(A.2)} \end{aligned}$$

welches sich gemäß der Exponentialdarstellung aus einem Amplitudensignal $A(t)$ sowie dem gewünschten Phasensignal $\phi(t)$ zusammensetzt. Dabei bezeichnet $\hat{s}(t)$ die Hilbert-Transformierte des Signals $s(t)$, die sich mit Hilfe der Fourier-Transformation numerisch effizient berechnen lässt [34, 35]. Aus dem Signal und dessen Hilbert-Transformierten kann dementsprechend das Phasensignal

$$\phi(t) = arctan\left(\frac{\hat{s}(t)}{s(t)}\right)\,. \qquad \text{(A.3)}$$

bestimmt werden.

Liegt eine Kopplung zwischen zwei eigenständig schwingfähigen Signalen vor, die zu einer Synchronisation führt, so besteht zwischen den Phasen der beiden Signale eine Beziehung. Die Differenz der Phasen ist dann beschränkt [36], und entspricht der Bedingung

$$|m\phi^{(1)}(t) - n\phi^{(2)}(t)| < const\,. \tag{A.4}$$

Die Faktoren n, m berücksichtigen dabei, dass Synchronisation bei unterschiedlichen Frequenzen eintreten kann. Im Folgenden werden die Faktoren m und n auf 1 gesetzt.

Ein Maß für die Synchronisation zweier Signale ergibt sich aus der Verteilung der Phasendifferenz. Dabei wird nicht die Differenz selbst betrachtet, sondern in der Regel die Differenz modulo 2π

$$\psi(t) = (\phi^{(1)}(t) - \phi^{(2)}(t)) \bmod 2\pi\,, \tag{A.5}$$

die möglicherweise auftretenden Phasensprüngen Rechnung trägt. Im Falle perfekt synchronisierter Systeme ist die Phasendifferenz modulo 2π konstant, was zu einem scharfen Peak in der Verteilung der Phasendifferenz führt. Bei ungekoppelten Signalen ergibt sich hingegen eine Gleichverteilung der Phasen. Dies kann durch den normierten Phasensynchronisationsindex [18]

$$R^2 = \langle \cos\psi(t)\rangle^2 + \langle \sin\psi(t)\rangle^2 \tag{A.6}$$

quantifiziert werden. Er nimmt Werte nahe bei 0 an, falls keine Beziehung zwischen den Phasen vorliegt, und Werte bei 1, falls entsprechende Phasensynchronisation vorliegt.

Das besondere an der Phasensynchronisation ist dabei, dass sich ein hoher Phasensynchronisationsindex bereits bei sehr schwacher Kopplung einstellen kann. Bei solchen Kopplungen kann eine Synchronisation z. B. der Amplituden der Signale noch nicht nachgewiesen werden.

A.2 Dynamical Similarity Index

Beim „Dynamical Similarity Index“ wird die Dynamik eines gleitenden Datenfensters S_t mit der eines Referenzfensters S_{ref} verglichen [20, 37], wie in Abschnitt 2.3.2 angeführt. Zum Zweck der Vorhersage epileptischer Anfälle wird dabei aus den EEG-Daten ein Referenzfenster gewählt, das fern von Anfällen liegt, und möglichst frei von Anfallsaktivität ist.

Um die den Fenstern zugrundeliegende Dynamik zu rekonstruieren, werden die Zeitpunkte der positiven Nulldurchgänge T_n der Daten ermittelt, und die Zeitintervalle zwischen den Nulldurchgängen betrachtet, was zu einer neuen Zeitreihe $I_n = T_{n+1} - T_n$ führt. Aus dieser werden über eine Einbettung der Dimension $m = 15$ die Einbettungsvektoren $A_n = (I_n, I_{n-1}, ..., I_{n-m-1})$ bestimmt [38]. Zur Reduktion des Rauschens wird eine Singulärwertzerlegung der Trajektorienmatrizen $\mathbf{A}(S_{ref})$ bzw. $\mathbf{A}(S_t)$ angewendet. Diese werden anschließend auf die vier größten Hauptachsen des Referenzfensters projiziert, wodurch man $\mathbf{X}(S_t)$ bzw. $\mathbf{X}(S_{ref})$ erhält.

Da das Referenzfenster normalerweise mit einer Länge von fünf Minuten gewählt wird, während für das gleitende Fenster eine Länge von 25 Sekunden gewählt wurde, wird eine zufällig eine Teilmenge $\mathbf{Y}(S_{ref})$ von $\mathbf{X}(S_{ref})$ mit $N_{ref} \approx N_t$ Punkten für den Vergleich der Fenster bestimmt. Als ein Maß der Ähnlichkeit beider Fenster wird das Kreuzkorrelationsintegral

$$C(S_{ref}, S_t) = \frac{1}{N_{ref} N_t} \sum_{i=1,N_{ref}} \sum_{j=1,N_t} \Theta\left(r - ||\mathbf{Y}_i(S_{ref}) - \mathbf{X}_j(S_t)||\right) \tag{A.7}$$

berechnet [39], wobei $||\cdot||$ die Euklidische Norm ist. Die Heaviside-Funktion Θ liefert hier den Wert 1 zurück, falls die Punkte $\mathbf{Y}_i(S_{ref})$ und $\mathbf{X}_j(S_t)$ in einem Abstand geringerer als r liegen, ansonsten 0. Standardmäßig wird der Abstand so gewählt, dass sich im Mittel von jedem beliebigen Punkt des Referenzfensters mindestens 30% aller Punkte des Referenzfensters in einem Abstand von höchstens r befinden. Zur Normierung auf Werte im Bereich $[0, 1]$ ist der „Dynamical Similarity Index“ definiert als

$$\gamma(S_t) = \frac{C(S_{ref}, S_t)}{\sqrt{C(S_{ref}, S_{ref}) C(S_t, S_r)}} \,. \tag{A.8}$$

Falls die zugrundeliegende Dynamik des Referenzfensters S_{ref} und des gleitenden Fensters S_t sehr ähnlich ist, ergibt sich ein Wert nahe 1, und bei veränderter Dynamik des gleitenden Fensters ein Wert kleiner als 1.

A.3 Random Predictor

Zum Vergleich der vorgestellten Vorhersagemethoden mit mit einer unspezifischen Methode, welche Alarme zufällig auslöst, wurde der „Random Predictor“ verwendet [21, 24]. Wie auch den spezifischen Methoden wird diesem erlaubt, innerhalb eines kurzen Zeitfensters I Alarme entsprechend der maximalen Fehlerrate $FPR_{\max}$ auszulösen, so dass $p = FPR_{\max} \cdot I$ die Wahrscheinlichkeit für einen Alarm während des Zeitfensters I ist. Um einen Anfall korrekt vorherzusagen, muss mindestens ein Alarm während eines längeren Zeitraums SOP ausgelöst werden. Die Wahrscheinlichkeit hierfür ist

$$
\begin{aligned}
P(\text{mind. ein Alarm in } SOP) &= 1 - P(\text{kein Alarm in } SOP) && \text{(A.9)}\\
&= 1 - P(\text{kein Alarm in I})^{\frac{SOP}{I}} && \text{(A.10)}\\
&= 1 - (1 - FPR_{\max} \cdot SOP)^{\frac{SOP}{I}} && \text{(A.11)}\\
&\approx 1 - e^{-FPR_{\max} \cdot SOP} \text{ für } I \ll SOP && \text{(A.12)}
\end{aligned}
$$

Die Wahrscheinlichkeit P, einen Anfall korrekt vorherzusagen, hängt nur von den Vorhersageparametern SOP und $FPR_{\max}$ ab. Sie kann verwendet werden, um gemäß einer Binomialverteilung die Wahrscheinlichkeit zu berechnen, von K eingetretenen Anfällen mindestens k vorherzusagen

$$
P_{\text{binom}}\{k; K; P\} = \sum_{j \geq k} \left(\begin{array}{c} K \\ j \end{array} \right) P^j (1 - P)^{K-j} \; . \qquad \text{(A.13)}
$$

Wie in 3.1.2.1 besprochen werden im Zuge der Optimierung der spezifischen Vorhersagemethoden mehrfach beste Parameter bestimmt, z. B. das Beste von ursprünglich mehreren Features. Um einen adäquaten Vergleich zu erlauben, muss auch dem „Random Predictor“ diese mehrmalige Gelegenheit eingeräumt werden, durch korrekte Alarme die Anfälle „vorherzusagen“. Unter der Annahme, dass der spezifischen Methode d freie Parameter zur Verfügung stehen, ergibt dies dementsprechend

$$P_{\text{binom},\,d}\left\{k;K;P\right\} = 1 - \left(\sum_{j<k} \left(\begin{array}{c} K \\ j \end{array} \right) P^j \left(1-P\right)^{K-j}\right)^d . \tag{A.14}$$

Damit steht eine Möglichkeit zur Verfügung, um abzuschätzen, wie viele der beobachteten Anfälle mit einer hinreichenden Wahrscheinlichkeit auch von einer zufälligen Vorhersagemethode korrekt hätten vorhergesagt werden können.

Anhang B

Verwendete *XML*-Formate

B.1 Konfiguration des Analyse-Projektes

```
<!DOCTYPE Project [
  <!ELEMENT Project (ResultsPath,InfoPath,DataSources,DataProcessors,
                     FPRmaxs,SPHs,SOPs,FPRmaxs)>
      <!ATTLIST Project type (Optimization|Testing|Online) #REQUIRED>
      <!ATTLIST Project maximumBlockDistance CDATA #REQUIRED>
      <!ATTLIST Project maximumGap CDATA #REQUIRED>
      <!ATTLIST Project maximumBlockLength CDATA #REQUIRED>
      <!ATTLIST Project postIctalPeriod CDATA #REQUIRED>
  <!ELEMENT InfoPath    (#PCDATA)>
  <!ELEMENT ResultsPath    (#PCDATA)>
  <!ELEMENT DataSources (DataSource+)>
  <!ELEMENT DataSource (FilesPath,Thresholds)>
      <!ATTLIST DataSource type CDATA #REQUIRED>
      <!ATTLIST DataSource name CDATA #REQUIRED>
  <!ELEMENT FilesPath    (#PCDATA)>
  <!ELEMENT Thresholds    (#PCDATA)>
      <!ATTLIST Thresholds direction (positive|negative) #REQUIRED>
      <!ATTLIST Thresholds stayAboveBelowTime CDATA #REQUIRED>
  <!ELEMENT DataProcessors (DataProcessor*)>
  <!ELEMENT DataProcessor (Parameter*)>
      <!ATTLIST DataProcessor name CDATA #REQUIRED>
  <!ELEMENT Parameter (#PCDATA)>
    <!ATTLIST Parameter name CDATA #REQUIRED>
  <!ELEMENT FPRmaxs (#PCDATA)>
  <!ELEMENT SOPs (#PCDATA)>
  <!ELEMENT SPHs (#PCDATA)>
]>
```

Auflistung B.1: Definition des Dokumententyps zur Projekt-Konfiguration.

B.2 Konfiguration der Patienten-Informationen

```
<!ELEMENT Patients (Patient+)>
<!ELEMENT Patient (Subfolder, FeatureFiles, ResultsSubfolder, IncludePeriods,
    ExcludePeriods, RealSeizuresFile)>
  <!ATTLIST Patient number CDATA #REQUIRED>

<!ELEMENT Subfolder (#PCDATA)>
<!ELEMENT FeatureFiles (#PCDATA)>
  <!ATTLIST FeatureFiles type CDATA #REQUIRED>
<!ELEMENT ResultsSubfolder (#PCDATA)>

<!ELEMENT IncludePeriods (Period)>
  <!ATTLIST IncludePeriods format CDATA #REQUIRED>
<!ELEMENT IncludePeriods (Period)>
  <!ATTLIST ExcludePeriods format CDATA #REQUIRED>

<!ELEMENT RealSeizuresFile (#PCDATA)>
  <!ATTLIST RealSeizuresFile format CDATA #REQUIRED>

<!ELEMENT Period (Start, End)>
<!ELEMENT Start (#PCDATA)>
<!ELEMENT End (#PCDATA)>
```

Auflistung B.2: Definition des Dokumententyps zur Konfiguration der Patienten-Informationen.

B.3 Konfiguration der Datenquellen

```
<!ELEMENT DataSources (DataSource+)>
<!ELEMENT DataSource (Feature+)>
  <!ATTLIST DataSource name CDATA #REQUIRED>
  <!ATTLIST DataSource type CDATA #REQUIRED>

<!ELEMENT Feature (FileExtension, Abbreviation)>
  <!ATTLIST DataSource minValue CDATA #REQUIRED>
  <!ATTLIST DataSource maxValue CDATA #REQUIRED>
  <!ATTLIST DataSource columnsAfter CDATA #REQUIRED>
  <!ATTLIST DataSource columnsBefore CDATA #REQUIRED>

<!ELEMENT FileExtension (#PCDATA)>
<!ELEMENT Abbreviation (#PCDATA)>
```

Auflistung B.3: Definition des Dokumententyps zur Datenquellen-Konfiguration.

B.4 Ausgabeformat für die Vorhersage-Ergebnisse

```
1  <!ELEMENT PredictionResults ( PredictionResult* ) >
2  <!ELEMENT PredictionResult ( UsedSeizures, InterIctalTime, Feature,
       Threshold, Corrects, Incorrects, FPR, Sensitivity, IncorrectAlarms*,
       CorrectAlarms* ) >
3      <!ATTLIST PredictionResult FPRmax #REQUIRED >
4      <!ATTLIST PredictionResult SOP #REQUIRED >
5      <!ATTLIST PredictionResult SPH #REQUIRED >
6
7  <!ELEMENT UsedSeizures ( #PCDATA ) >
8  <!ELEMENT InterIctalTime ( #PCDATA ) >
9  <!ELEMENT Feature ( #PCDATA ) >
10 <!ELEMENT Threshold ( #PCDATA ) >
11 <!ELEMENT FPR ( #PCDATA ) >
12 <!ELEMENT Sensitivity ( #PCDATA ) >
13
14 <!ELEMENT CorrectAlarms ( #PCDATA ) >
15 <!ELEMENT Corrects ( #PCDATA ) >
16 <!ELEMENT IncorrectAlarms ( #PCDATA ) >
17 <!ELEMENT Incorrects ( #PCDATA ) >
```

Auflistung B.4: Definition des Dokumententyps für die Ausgabe der Vorhersage-Ergebnisse.

Anhang C

Quellcode des PredictionSystem

```
1  #if !defined PREDICTION_SYSTEM_H
2  #define PREDICTION_SYSTEM_H
3
4  #include "AnalysisFramework.h"
5
6  using namespace std;
7
8  class PredictionSystem;
9  class Parameter;
10
11 ////////////////////////////////////////////////////////////////////////
12 ///  PredictionSystem
13 ///  Class modelling the prediction system as a state machine
14 ///  with regards to a specific prediction characteristic (SPH,SOP,FPRmax)
15 ///
16 ///  @remarks uses project wide settings defined in AnalysisProject
17 ///
18 ////////////////////////////////////////////////////////////////////////
19 class PredictionSystem
20 {
21 public:
22   PredictionSystem();
23
24   void SetParameter(Parameter* para) { parameter=para; }
25
26   void SignalAlarm(time_t alarmTime);
27   void SignalSeizure(TimePeriod seizure);
28
29   void SetLastSeizure(int lastSeizure) { this->lastSeizure=lastSeizure; }
30   int  GetLastSeizure() { return this->lastSeizure; }
31
32 protected:
33
```

```
34    void CheckTimeouts(time_t time);
35
36    ///  PredictionState: enum of available states
37    enum PredictionState
38    {
39      InterIctal,            ///< Inter ictal state
40      PreIctalSPH,           ///< Pre ictal state during SPH
41      PreIctalSOP,           ///< Pre ictal state during SOP
42      PostIctal              ///< Post ictal state
43    };
44
45    PredictionState state;   ///< Current system state
46    time_t currentTime;      ///< Current time (in seconds)
47    time_t lastAlarmTime;    ///< Time of the last alarm (in seconds)
48    time_t lastSeizureTime;  ///< Time of the last seizure (in seconds)
49    int    lastSeizure;         ///< No. of the last seizure
50
51    Parameter* parameter;    ///< Prediction parameters (stored externally)
52  };
53
54  #endif // PREDICTION_SYSTEM_H
```

Auflistung C.1: Header-Datei der Klasse PredictionSystem.

```
1   #include "AnalysisFramework.h"
2
3
4   ////////////////////////////////////////////////////////////////////////
5   ///  public constructor  PredictionSystem
6   ///
7   PredictionSystem::PredictionSystem() {
8
9     state = InterIctal;
10    lastAlarmTime = -1;
11    lastSeizureTime = -1;
12    lastSeizure = -1;
13  }
14
15  ////////////////////////////////////////////////////////////////////////
16  ///  public  SignalAlarm
17  ///  Signals an alarm event
18  ///
19  ///  @param [in]       alarmTime time_t    Time of the alarm (in seconds)
20  ///
21  void PredictionSystem::SignalAlarm(time_t alarmTime) {
22
23    CheckTimeouts(alarmTime);
24
```

```
25    if(state == InterIctal) {
26      state = PreIctalSPH;
27      lastAlarmTime = alarmTime;
28    }
29
30  }
31
32
33  ////////////////////////////////////////////////////////////////////////
34  ///  public  SignalSeizure
35  ///  Signals a seizure
36  ///
37  ///  @param [in]       seizure TimePeriod     Time of the seizure (in seconds)
38  ///
39  void PredictionSystem::SignalSeizure(TimePeriod seizure) {
40
41    CheckTimeouts(seizure.startTime);
42
43    if(state == InterIctal) {
44
45      // unpredicted seizure
46      state = PostIctal;
47    } else if(state == PreIctalSPH) {
48
49      // incorrect alarm
50      parameter->AddIncorrectAlarm(lastAlarmTime);
51      state = PostIctal;
52
53    } else if(state == PreIctalSOP) {
54      }
55
56      // correct prediction
57      parameter->AddCorrectAlarm(lastAlarmTime);
58
59      state = PostIctal;
60      currentTime = seizure.endTime + project.timeAfterSeizure;
61
62    } else if(state == PostIctal) {
63      // ignore
64    }
65
66    lastSeizureTime = seizure.endTime;
67  }
68
69
70  ////////////////////////////////////////////////////////////////////////
71  ///  public  CheckTimeouts
72  ///  Method to regularly check possible timeouts of the prediction system
```

```
73 ///
74 ///  @param [in]       time time_t     Time to check timeouts (in seconds)
75 ///
76 void PredictionSystem::CheckTimeouts(time_t time) {
77
78   if(state == PreIctalSPH) {
79     if(time > lastAlarmTime + parameter->SPH) {
80       state = PreIctalSOP;
81     }
82   }
83
84   if(state == PreIctalSOP) {
85
86     if(time > lastAlarmTime + parameter->SPH + parameter->SOP) {
87       state = InterIctal;
88       parameter->incorrects.push_back(lastAlarmTime);
89     }
90   }
91
92   if(state == PostIctal) {
93     if(time > lastSeizureTime + project.timeAfterSeizure) {
94       state = InterIctal;
95     }
96   }
97
98 }
```

Auflistung C.2: Implementation der Klasse PredictionSystem.

Anhang D

Technische Details der Implementation

Der Entwurf des Programms „SeizureAnalyzer“ wurde mittels der objektorientierten Programmiersprache *C++* implementiert. Dabei wurde auf weitgehende Quellcode-Portabilität geachtet, so dass das Programm unter den Betriebssystemen *GNU/Linux* sowie *Microsoft Windows* kompiliert und eingesetzt werden kann. Es wurden Projektdateien für die *GNU Make*-Umgebung zur Verfügung gestellt [40]. Dokumentiert wurde der Quellcode mithilfe des Dokumentationswerkzeugs „Doxygen“ [41].

Zum Parsen der verwendeten *XML*-Formate wurde die *XML*-Bibliothek *libxml* [42] verwendet, welche über den *C++*-Wrapper *libxml++* eingebunden wurde. Für die Berechnung des laufenden Medians durch die Klasse MedianSmoother wurde auf die Template-Klasse sorted_vector zurückgegriffen [43], welche eine Multimenge mit wahlfreien Zugriff auf einzelne Elemente zur Verfügung stellt.

Literaturverzeichnis

[1] J.S. Ebersole: *In search of seizure prediction: a critique.* Journal of Clinical Neurophysiology, 116:489–92, 2005.

[2] C.E. Elger: *Future trends in epileptology.* Current Opinion in Neurology, 14:185–186, 2001.

[3] A. Schulze-Bonhage, C. Kurth, A. Carius, B.J. Steinhoff, and T. Mayer: *Seizure anticipation by patients with focal and generalized epilepsy: A multicentre assessment of premonitory symptoms.* Epilepsy Research, 2006.

[4] F. Mormann, C.E. Elger, and K. Lehnertz: *Seizure anticipation: from algorithms to clinical practice.* Current Opinion in Neurology, 19:187–193, 2006.

[5] R.S. Fisher, W. van Emde Boas, W. Blume, C. Elger, P. Genton, P. Lee, and J. Engel: *Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE).* Epilepsia, 46:470–472, 2005.

[6] Deutsche Sektion der Internationalen Liga gegen Epilepsie: *Mitteilungen.* Zeitschrift für Epileptologie, 16:264–276, 2003.

[7] R.S. Fisher. American Epilepsy Society News, 5, 1996.

[8] H. Stefan, F.H. Lopes da Silva, W. Löscher, D. Schmidt, E. Perucca, M. J. Brodie, P. A J M Boon, W. H. Theodore, and S. L. Moshé: *Epileptogenesis and rational therapeutic strategies.* Acta Neurologica Scandinavica, 113:139–155, 2006.

[9] S. Wiebe, W.T. Blume, J.P. Girvin, and M. Eliasziw: *A randomized, controlled trial of surgery for temporal-lobe epilepsy.* The New England Journal of Medicine, 345:311–8, 2001.

[10] W.H. Theodore and R.S. Fisher: *Brain stimulation for epilepsy.* The Lancet Neurology, 3:111–118, 2004.

[11] D. M. Andrade, D. Zumsteg, C. Hamani, M. Hodaie, S. Sarkissian, A. M. Lozano, and R. A. Wennberg: *Long-term follow-up of patients with thalamic deep brain stimulation for epilepsy.* Neurology, 2006.

[12] J. Martinerie, C. Adam, M. Le van Quyen, M. Baulac, S. Clemenceau, B. Renault, and F.J. Varela: *Epileptic seizures can be anticipated by non-linear analysis.* Nature Medicine, 4:1173–1176, 1998.

[13] K. Lehnertz, R. G. Andrzejak, J. Arnhold, T. Kreuz, F. Mormann, C. Rieke, G. Widman, , and C.E. Elger: *Nonlinear EEG analysis in epilepsy: its possible use for interictal focus localization, seizure anticipation, and prevention.* Journal of Clinical Neurophysiology, 18:209–222, 2001.

[14] P.L. Nunez and R. Srinivasan: *Electric Fields of the Brain – The Neurophysics of EEG.* Oxford University Press, second edition, 2006.

[15] D. Denny-Brown: *The electro-encephalogram in epilepsy.* Epilepsia, 1:173–240, 1939.

[16] D. Kovalev, J. Spreer, J. Honegger, J. Zentner, A. Schulze-Bonhage, and H. J. Huppertz: *Rapid and fully automated visualization of subdural electrodes in the presurgical evaluation of epilepsy patients.* American Journal of Neuroradiology, 26:1078–1083, 2005.

[17] B. Litt and J. Echauz: *Prediction of epileptic seizures.* Lancet Neurology, 1:22–30, 2002.

[18] F. Mormann, K. Lehnertz, P. David, and C.E. Elger: *Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients.* Physica D, 144:358–369, 2000.

[19] F. Mormann, R.G. Andrzejak, T. Kreuz, C. Rieke, P. David, C.E. Elger, and K. Lehnertz: *Automated detection of a pre-seizure state based on a decrease in synchronization in intracranial EEG recordings from epilepsy patient.* Physical Review E, 67:021912, 2003.

[20] M. Le van Quyen, J. Martinerie, M. Baulac, and F. Varela: *Anticipating epileptic seizures in real time by a non-linear analysis of similarity between EEG record-*

ings. Neuroreport, 10:2149–2155, 1999.

[21] M. Winterhalder, T. Maiwald, H. U. Voss, R. Aschenbrenner-Scheibe, J. Timmer, and A. Schulze-Bonhage: *The seizure prediction characteristic: A general framework to assess and compare seizure prediction methods.* Epilepsy & Behavior, 4:318–325, 2003.

[22] R. Sowa, A. Chernihovskyi, F. Mormann, and K. Lehnertz: *Estimating phase synchronization in dynamical systems using cellular nonlinear networks.* Physical Review E, 71:061926, 2005.

[23] T. Maiwald, M. Winterhalder, R. Aschenbrenner-Scheibe, H.U. Voss, A. Schulze-Bonhage, and J. Timmer: *Comparison of three nonlinear seizure prediction methods by means of the seizure prediction characteristic.* Physica D, 194:357–368, 2004.

[24] B. Schelter, M. Winterhalder, T. Maiwald, A. Brandt, A. Schad, A. Schulze-Bonhage, and J. Timmer: *Testing statistical significance of multivariate time series analysis techniques for epileptic seizure prediction.* Chaos, 16:013108, 2006.

[25] M. Born, E. Holz und O. Kath: *Softwareentwicklung mit UML 2.* Addison-Wesley, 2004.

[26] Object Management Group: *Unified Modeling Language (UML), version 2.0.* `http://www.omg.org/technology/documents/formal/uml.htm`, 2005.

[27] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau: *Extensible markup language (XML) 1.0 (third edition) – W3C recommendation 04 February 2004.* `http://www.w3.org/TR/2004/REC-xml-20040204/`, 2004.

[28] A. Müller, H. Osterhage, R. Sowa, R.G. Andrzejak, F. Mormann, and K. Lehnertz: *A distributed computing system for multivariate time series analyses of multichannel neurophysiological data.* Journal of Neuroscience Methods, 152:190–201, 2006.

[29] P.J. Durka and D. Ircha: *SignalML: metaformat for description of biomedical time series.* Computer Methods and Programs in Biomedicine, 76:253–259, 2004.

[30] S. Mintert und H. Behme: *Schlüsselqualifikation – XML jenseits des Mainstream.* iX – Magazin für professionelle Informationstechnik, 2005.

[31] K.N. Fountas, J.R. Smith, A.M. Murro, J. Politsky, Y.D. Park, and P.D. Jenkins: *Implantation of a closed-loop stimulation in the management of medically refractory focal epilepsy: a technical note.* Stereotactic and Functional Neurosurgery, 83:153–158, 2005.

[32] J. Almeida: *XML4MAT*, 2004. `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6268`.

[33] J. Engel: *Outcome with respect to epileptic seizures.* In J. Engel (editor): *Surgical Treatment of the Epilepsies*, pages 553–71, New York, 1987. Raven Press.

[34] A.V. Oppenheim and R.W. Schafer: *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs AND NJ, 1975.

[35] M. Winterhalder: *Analyse der Synchronisation multivariater Systeme.* Diplomarbeit, Albert-Ludwigs-Universität Freiburg, 2003.

[36] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths: *Phase synchronization of chaotic oscillators.* Physical Review Letters, 76:1804–1807, 1996.

[37] M. Le van Quyen, C. Adam, J. Martinerie, M. Baulac, S. Clemenceau, and F. Varela: *Spatio-temporal characterizations of non-linear changes in intracranial activities prior to human temporal lobe seizures.* European Journal of Neuroscience, 12:2124–2134, 2000.

[38] T. Sauer: *Reconstruction of dynamical systems from interspike intervals.* Physical Review Letters, 72:3821–3814, 1994.

[39] H. Kantz: *Quantifying the closeness of fractal measures.* Physical Review E, 49:5091–5097, 1994.

[40] R. M. Stallman, R. McGrath, and P. D. Smith: *GNU Make: A Program for Directing Recompilation.* Free Software Foundation, 2004.

[41] D. van Heesch: *Doxygen.* `http://www.stack.nl/~dimitri/doxygen`.

[42] *LibXML – The XML C parser and toolkit of Gnome.* `http://www.xmlsoft.org`.

[43] M. Holzherr: *An STL compliant sorted vector.* `http://www.codeproject.com/vcpp/stl/sorted_vector.asp`.

Danksagung

Allen, die an der Entstehung dieser Diplomarbeit beteiligt waren, sei an dieser Stelle gedankt. Vielfacher Dank gilt Herrn Prof. Dr. Hans Jürgen Ohlbach für seine Betreuung dieser Arbeit.

Gleichfalls sehr herzlich danke ich Herrn Prof. Dr. Jens Timmer für die Überlassung des Themas und die Betreuung und Unterstützung dieser interdisziplinären Arbeit im Rahmen seiner Freiburger Arbeitsgruppe. Insbesondere Matthias Winterhalder und Björn Schelter haben es mir in hilfreichen Diskussionen sowie durch sehr konstruktive Kritik und Ratschläge ermöglicht, einen erweiterten Zugang zu den zugrundeliegenden wissenschaftlichen Fragestellungen zu finden. Auch für die angenehme Arbeitsatmosphäre gilt ihnen mein herzlicher Dank.

Die durchgeführten Untersuchungen der Vorhersage epileptischer Anfälle waren durch eine Kooperation mit der Sektion für Prächirurgische Epilepsiediagnostik des Neurozentrums der Universität Freiburg möglich. Für die Bereitstellung der EEG-Daten wie auch für die Unterstützung in medizinischen Fragen danke ich Herrn Privatdozent Dr. Andreas Schulze-Bonhage sehr. Allen medizinisch-technischen Assistenten gilt mein herzlicher Dank für die wertvolle Aufbereitung der Daten, und die entgegenkommende Zusammenarbeit – stellvertretend sei Carolin Gierschner genannt. Armin Brandt, Johannes Wohlmuth und Jakob Nawrath möchte ich ferner für ihre Unterstützung und Mitarbeit sehr danken.

Für das Korrekturlesen der Arbeit geht mein Dank an Björn Schelter, Matthias Winterhalder, Birgitta Feldwisch genannt Drentrup, Arndt Twickel und Andre Nuppenau.

Ein herzliches Dankeschön auch allen weiteren Menschen, deren Unterstützung ich im Studium und bei der Diplomarbeit erfahren durfte. Speziell meinen Eltern gebührt diesbezüglich viel Dank.