DISS. ETH NO. 27969

Motifs and Manifolds

Statistical and Topological Machine Learning for Characterising and Classifying Biomedical Time Series

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by

CHRISTIAN BOCK
M.Sc., Heidelberg University

born on 27.08.1990
citizen of Germany

accepted on the recommendation of

Prof. Dr. Karsten Borgwardt, examiner
Prof. Dr. Finale Doshi-Velez, co-examiner
Prof. Dr. Smita Krishnaswamy, co-examiner

2021

Meinen Eltern.

Abstract

The increased focus on evidence-based practice in the health sciences led to a plethora of (un)organised and digitised data. In conjunction with the availability of technological advances in the life sciences, this resulted in extraordinary access to biomedical data. Due to efficient measurement devices, the frequency at which data can be obtained is at an unprecedented high, leading to the adage that data, indeed, could be the new gold. Examples of such high-resolution time series data are the continuous monitoring of patient vital parameters or a single electrocardiogram (ECG) itself. The temporal component introduced by time series data is both a chance and a challenge, necessitating the development of appropriate data analysis techniques. A chance, as it allows us to utilise a measurement's temporal evolution to characterise or classify the object of interest (e.g. patients, cells, or other organisms). A challenge because local and global correlation structures exacerbate obtaining a complete picture of a time-evolving phenomenon. Moreover, many dynamically changing systems exhibit alterations that occur at multiple scales and in multiple channels.

This thesis presents a set of novel methods to help characterise and classify time-varying data with the express purpose of answering questions at the intersection of machine learning and healthcare. Recognising that time series arise from different categories, we first separate them into real-valued and object-valued time series and investigate both types separately.

For the analysis of the first type, we propose a novel method to mine time series patterns efficiently. Driven by a statistical approach, we will introduce a way to identify temporal biomarkers and illustrate their utility in a data set of intensive care unit patients. For this, we leverage the expressive power of subsequences to obtain a high-dimensional time series representation. This feature representation is subsequently used to develop a kernel method based on optimal transport theory. The developed algorithm is of general applicability for medium-sized data sets and has proven particularly effective in the classification setting. The first part of this thesis ends with the presentation of a collaborative machine learning system to predict myocardial ischaemia from stress test ECGs. We develop a deep learning-based approach to significantly reduce the number of patients that unnecessarily undergo myocardial perfusion imaging. A subsequent interpretability analysis presents a potential path towards explainable and trustworthy artificial intelligence in cardiology.

The second part of this thesis describes an effort to improve our understanding of artificial neural networks. By treating the network as a composition of time-varying graphs, we develop a method that characterises the change of its structural complexity over time. Our method captures the benefit of deep-learning best practices and can be used as an early-stopping criterion without the need for a validation data set. We thus manage to improve our

understanding of artificial neural networks and shed light on the properties linked to their generalisation capabilities.

Throughout this thesis, we demonstrate and highlight that in the analysis of (biomedical) time series, it is crucial to take the end-user into account. Interpretability and statistical analyses are of utter importance to make the otherwise opaque field of machine learning transparent to clinicians, physicians, and biologists. Moreover, we also hold up the mirror to ourselves as machine learning researchers: Comprehending the underlying mechanisms of our algorithms is at least as important as their empirical successes. The present thesis paves the path towards a better understanding of artificial neural networks and sheds light on complex phenotypes such as sepsis and myocardial ischaemia in clinically relevant ways.

Zusammenfassung

Die zunehmende Konzentration auf die evidenzbasierte Praxis in der Medizin führt zu einer Fülle von (un)organisierten und digitalen Daten. In Verbindung mit technischen Fortschritten in der Biologie resultiert dies in einer außerordentlichen Flut an biomedizinischen Daten. Dank neuer und effizienter Messgeräte können Forscher und Kliniker in nie dagewesener Geschwindigkeit Daten gewinnen, was uns anregt, zu hinterfragen, ob Daten in der Tat das neue Gold sein könnten. Beispiele für solche hochauflösenden Zeitreihendaten sind die kontinuierliche Überwachung der Vitalparameter von Patient:innen auf der Intensivstation oder ihre Elektrokardiogramme (EKG). Die zeitliche Komponente, die Zeitreihendaten innewohnt, ist Chance und Herausforderung zugleich, erfordert sie doch die Entwicklung geeigneter Datenanalysetechniken. Sie stellt eine Chance dar, denn sie ermöglicht es uns, die zeitliche Entwicklung einer Messung zu untersuchen und zu nutzen, um das betreffende Objekt zu charakterisieren oder zu klassifizieren (z. B. Patienten, Zellen oder andere Organismen). Gleichzeitig ist sie auch eine Herausforderung, denn lokale und globale Korrelationsstrukturen erschweren es, ein vollständiges Bild von sich zeitlich entwickelnden Phänomenen zu erhalten. Darüberhinaus weisen viele sich dynamisch verändernden Systeme Veränderungen auf, die auf mehreren Skalen und in mehreren Kanälen stattfinden.

In dieser Dissertation stellen wir neuartige Methoden zur Charakterisierung und Klassifizierung von Zeitreihen vor, mit dem ausdrücklichen Ziel, Fragen an der Schnittstelle des maschinellen Lernens und Gesundheitswesens zu beantworten. Unserem Verständnis nach gehören Zeitreihen unterschiedlichen Kategorien an. Aus diesem Grund werden wir im Folgenden zwischen reellwertigen und abstrakten Zeitreihen unterscheiden und diese getrennt voneinander untersuchen.

Für die Analyse von Zeitreihen des ersten Typs untersuchen wir Aspekte der Charakterisierung und Klassifizierung von biomedizinischen Zeitreihen. Dafür stellen wir eine neue Methode zur effizienten Suche von Zeitreihenmotiven vor. Auf der Grundlage eines statistischen Ansatzes werden wir eine neuartige Methode zur Identifizierung temporaler Biomarker vorstellen und ihren Nutzen anhand eines Datensatzes von Intensivpflegepatient:innen veranschaulichen. Zu diesem Zweck werden wir die Ausdruckskraft von Teilsequenzen nutzen, die als hochdimensionale Repräsentation der Zeitreihe dienen werden. Diese Darstellung wird anschließend zur Entwicklung einer neuen Kernel-Methode genutzt, die ihre theoretischen Grundlagen aus der Transporttheorie gewinnt. Der von uns entwickelte Algorithmus ist allgemein nutzbar, besonders für mittelgroße Datensätze geeignet und hat sich vor allem in der Zeitreihenklassifizierung bewährt. Der erste Teil dieser Dissertation endet mit der Vorstellung eines Systems des kollaborativen maschinellen Lernens zur Vorhersage von Myokardischämie anhand von Daten, die während eines Belastungs-EKGs erhoben wurden.

Wir entwickeln einen auf Deep Learning basierenden Ansatz, um die Zahl der Behandelten, die sich unnötigerweise der kostspieligen und invasiven Myokardperfusionsbildgebung unterziehen, signifikant zu reduzieren. Durch die Analyse von Interpretierbarkeit heben wir den Nutzen von Deep Learning für die medizinische Zeitreihenanalyse hervor und zeigen einen möglichen Weg zu erklärbarer und vertrauenswürdiger künstlicher Intelligenz in der Kardiologie auf.

Der zweite Teil dieser Dissertation zielt darauf ab, ein besseres Verständnis von künstlichen neuronalen Netzen zu gewinnen; ein Anliegen, dessen Behandlung in gewisser Weise noch in den Kinderschuhen steckt. Wir untersuchen, wie sich künstliche neuronale Netze während des Trainings verändern. Indem wir das Netzwerk als eine Zusammensetzung von Graphen betrachten, entwickeln wir eine Methode, welche die Veränderung der strukturellen Komplexität über die Zeit hinweg charakterisiert. Auf diese Weise können wir unser Verständnis von künstlichen neuronalen Netzen verbessern und beginnen, ihre Generalisierungseigenschaften besser zu erläutern.

Insgesamt, wird in dieser Dissertation aufgezeigt und hervorgehoben, dass es bei der Analyse von (biomedizinischen) Zeitreihen entscheidend ist, den Benutzer zu berücksichtigen. Interpretierbarkeit und statistische Analysen sind von größtmöglicher Bedeutung, um ansonsten undurchsichtige Verfahren des maschinellen Lernens für Kliniker:innen und Biolog:innen transparenter zu machen. Des Weiteren halten wir uns als Forscher im Bereich des maschinellen Lernens selbst den Spiegel vor: Das Verständnis der Mechanismen, die unseren Algorithmen zugrunde liegen, ist mindestens genauso wichtig wie ihre empirischen Erfolge. Die vorliegende Dissertation ebnet nicht nur den Weg zu einem besseren Verständnis künstlicher neuronaler Netze, sondern beleuchtet auf klinisch relevante Weise auch komplexe Pathologien wie beispielsweise Sepsis und Myokardischämie.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Dr. Karsten Borgwardt, for providing me with the opportunity to pursue my doctoral degree in his group. I am thankful for his excellent scientific input, guidance and for creating an outstanding and challenging research environment in which I grew as a scientist and as a person. It is due to him that I became part of a community of exceptional individuals whose goal is to contribute to society by revolutionising healthcare and biomedicine.

Moreover, I would like to extend my thanks to Prof. Dr. Finale Doshi-Velez and Prof. Dr. Smita Krishnaswamy for their support as members of my doctoral committee and to Prof. Dr. Randall Platt for chairing my doctoral examination.

I am eternally grateful to my friend and mentor, Dr. Bastian Rieck. He provided me with a compass in the early days of my doctoral endeavours and always made sure I was using it appropriately. He went above and beyond as a caring mentor, dedicated teacher, and diligent co-author, helping me to see setbacks as opportunities. It is hard to put in words how thankful I am that our paths crossed and for his impeccable mentorship. Thank you for the many philosophical and technical discussions that broadened my horizon both academically and personally.

I was also incredibly fortunate to have had the chance to work with many brilliant co-authors. Thanks to Thomas Gumbsch, who was a role model in time management, and who contributed crucially to the very first publication of my doctoral studies. I would like to thank Dr. Michael Moor for being a curious colleague, driven co-author, and sincere friend. Thank you for the unforgettable table tennis sessions on the weekends, which always turned into in-depth discussions about our latest and future projects. I would also like to thank Max Horn for many discussions about the newest deep learning architectures and implementations, for hosting memorable dinners, and for being a reliable stronghold in Zurich. I am greatly indebted to Dr. Matteo Togninalli for every single brainstorming session that preceded both our shared principal authorships, for sharing the appetite to build things, and for great memories inside and outside the lab. Moreover, I wish to thank Dr. Damián Roqueiro for his support during my first year as a doctoral student and for the efforts he, Max, and Bastian directed towards providing a pristine computing infrastructure. Thanks to my colleagues and co-authors, Dr. Elisabetta Ghisu, for insightful discussions about kernel machines and to Dr. Catherine Jutzeler for her support. Thank you, Dr. Anja Gumpinger, for insightful technical discussions about statistics and the intricacies of hypothesis testing, and thanks to Leslie O'Bray for proofreading parts of this dissertation. I wish to thank Caroline Weis, Giulia Muzio, Dr. Katharina Heinrich, and Lucie Bourguignon for being marvellous colleagues; before and during the pandemic.

Having had the chance to be part of an impressive research group, I would also like to thank all present and past members of Prof. Karsten Borgwardt's lab, including Dr. Xiao He, Dr. Dean Bodenham, Dr. Daisuke Yoneoka, Dr. Lukas Folkman, Dr. Laetitia Papaxanthos, Dr. Llinares-López, Dr. Michael Adamer, Dr. Juliane Klatt, Dr. Sarah Brueningk, Tim Kucera, Bowen Fan, and Dr. Dexiong Chen. Moreover, I was lucky enough to work with great collaborators and would like to thank Prof. Dr. Christian Müller, Dr. Joan Walter, and Ivo Strebel. I want to also express my heartfelt thanks to everyone in the Biosystems Science and Engineering department who helped make my experience joyful. This includes Simon Höllerer and Dr. Philipp Koch for taking care of the physical health of the department and Cindy Malnasi for her exceptional administrative and organisational support.

I cannot thank Prof. Dr. Thomas Wetter enough for supporting me during my undergraduate and graduate studies at Heidelberg University. It was due to him that I was able to study abroad, gaining invaluable international research experience. Thanks to Prof. Dr. George Demiris, who gave me the chance to author my bachelor's thesis in his research group. Similarly, I would like to thank Prof. Dr. Sean Mooney for having been open to accepting me as a visiting student to write my master's thesis in his lab.

It goes without saying that I am forever grateful for all my supportive friends and family members. I wish to thank my brother Stefan who inspired me to study computer science and always supported me in good and challenging times. I am indebted to my parents, whose support is unconditional and filled with love and kindness. Last but certainly not least, thank you, Lea, for being there. Thank you for being my rock. Always.

Contents

1 Introduction

In which foundation and raison d'être of this thesis
are laid out.

Neither we [160] nor the data we generate can escape the all-encompassing influence of time. Any observation we make or measurement we take is unique in terms of its position on the time line. In contrast to static cross-sectional data, this temporal order characterises and defines a time series: it is a sequence of data points ordered by their creation time. In its most common instantiation, a time series either consists of scalar values (see Figure 1.1 for an example) or higher-dimensional vectors. Commonly, we refer to these data types as *univariate* and *multivariate* time series, respectively. However, time series can also consist of structured and complex objects such as images (i.e. videos) or graphs whose edge weights change over time.

Formally, let $\mathcal{P}$ be a space of time points, where $\mathcal{P} \subseteq \mathbb{R}$ or $\mathcal{P} \subseteq \mathbb{N}$. A time series is given by a map $T\colon \mathcal{P} \to \mathcal{Q}$ whose codomain $\mathcal{Q}$ determines its complexity. This allows us to define, increasingly ordered by their complexity, three important classes of time series.

Definition 1.1 (Class I: Univariate Time Series)**.** A univariate time series is given by a map

$$T\colon \mathcal{P} \to \mathcal{Q}, \text{ with } \mathcal{Q} = \mathbb{R}.$$

Definition 1.2 (Class II: Multivariate Time Series)**.** An n-dimensional multivariate time series is given by a map

$$T\colon \mathcal{P} \to \mathcal{Q}, \text{ with } \mathcal{Q} = \mathbb{R}^n.$$

Definition 1.3 (Class III: Object-valued Time Series)**.** An object-valued time series is given by a map

$$T\colon \mathcal{P} \to \mathcal{Q}, \text{ with } \mathcal{Q} = \{O_1, O_2, \ldots\},$$

where $\mathcal{Q}$ can be finite or countably infinite. O_i may be any structured object such as a graph, an image, or a cell.

Univariate time series exhibit the lowest complexity and the main challenge when analysing them is to infer temporal dependencies between individual observations. The second rung

of complexity subsumes the first and adds the problem of interaction between observations from different dimensions of the time series. Lastly, object-valued time series represent the most complex class adding questions about the structure of the object at hand. This may necessitate a certain set of analysis tools tailored specifically to the space $\mathcal{Q}$ of the objects. It is possible to reduce the complexity of class II time series to the one of univariate time series by neglecting any intra-dimensional interactions. For the remainder of this thesis, we will therefore refer to the first two classes as real-valued time series.

1.1 Real-Valued Time Series

In the life sciences, real-valued time series data is ubiquitous and a central data type in fields such as genomics [15] (as sequences of gene expression levels), neuroscience [39] (as spiking activity of neurons), and medicine [178] (as vital parameters of patients in the intensive care unit (ICU)), to name a few. In particular, biomedical databases such as the Medical Information Mart for Intensive Care (MIMIC) [122, 124] or the UK Biobank [251] are a sign of the unparalleled growth and importance of temporal data in biology and medicine. While from a reductionist's point of view, the data type itself may be defined as nothing more than an ordered sequence of values (or vectors), time series can exhibit temporal dependencies as complex as the underlying data generating process itself. This process defines the nature of the times series and its dynamics, however, many time series are composed of a symphony of more simple patterns. More specifically, we take the view that any sufficiently long time series may be decomposed into four simple patterns as illustrated in Figure 1.1:

- Trends
- Seasonality
- Cyclic variations
- Random fluctuations

Trends are the long-term mean tendencies of a time series that is independent of cyclic/irregular effects or the calendar. The long-term decrease in measles-related mortality among children as a result of vaccinations is an example of a healthcare-related long-term trend [269]. Seasonal patterns are calendar-related and systematic fluctuations occurring once or several times per year. For example, the amount of brown adipose fat tissue is higher in the winter months compared to the summer [290]. Cyclic variations refer to repeating patterns that are unrelated to the calendar. The duration of a cycle or its amplitude depends on the data source being analysed. Changes in hormone levels during a woman's menstrual cycle are

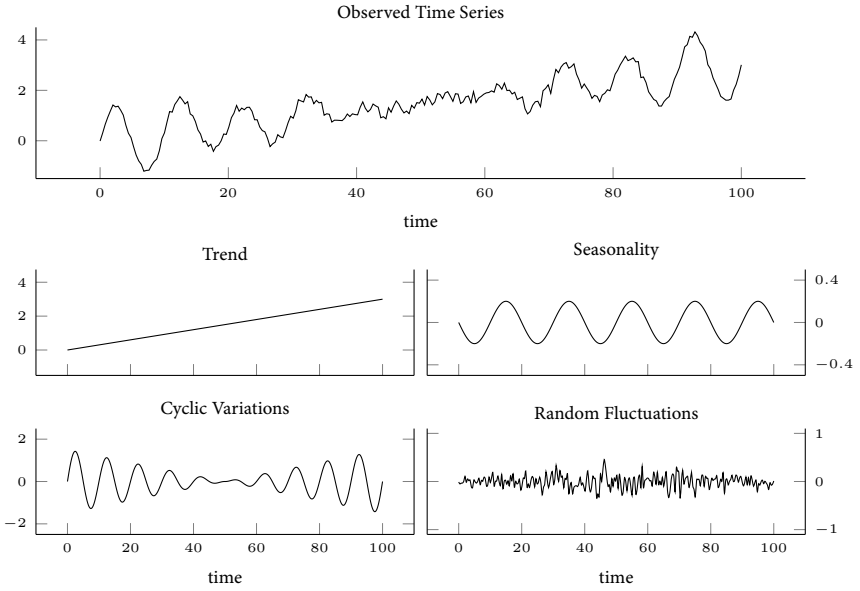


Figure 1.1: An observed time series may be composed of more simple individual patterns. In this case, the observed time series is the sum of a linear trend, seasonal and cyclic variations, as well as Gaussian noise.

an example of calendar-independent fluctuations [202]. Finally, random and uncontrollable behaviour such as noise induced by the measurement device or patient movements in the recording of electrocardiograms [183] are present in almost all biomedical time series and exacerbate their analysis.

While this view provides an intuitive description of the composition of a time series, formal properties unique to time series data exist. In the following paragraphs we will present important foundational concepts and outline our motivation to take a diverging path when analysing time series in this thesis.

1.1.1 Foundational Time Series Properties and Models

Many basic time series properties were developed in an econometric context [77] with the aim to predict future stock returns [267]. While financial time series differ from biomedical time series in many ways, most share the same fundamental properties and can, in theory, be analysed with the same methods. The key assumption of these approaches is that their central objects of interest are so-called linear time series.

Definition 1.4 (Linear Time Series)**.** Let μ be the mean of all observations v_t, where t indexes the time point at which an observation was made. Furthermore, let all a_t be independent and

identically distributed continuous random variables with zero mean and variance σ^2. Lastly, let Ψ_t be time-dependent coefficients. If v_t can be written as

$$v_t = \mu + \sum_{i=0}^{\infty} \Psi_i a_{t-i}, \tag{1.1}$$

we say the time series is *linear* [267, Equation 2.4].

For the modelling approaches we introduce in this section, we require the time series to be *weakly stationary*. In the context of a linear time series, this means that we have

$$\mathbb{E}[v_t] = \mu \qquad \text{and} \qquad \text{Cov}(v_t, v_{t-l}) = \gamma_l. \tag{1.2}$$

In other words, the mean of the time series is constant and the covariance between two observations is only a function of their temporal distance l. The latter is also referred to as the lag-l autocovariance of v_t [267], which brings us to one of the most fundamental properties of time series: serial dependence. Serial dependence means that an observation at time point t is statistically dependent on another observation from a different time point. This also means that altering the order of a time series will at least change the data's meaning (as it destroys the dependence structure) if not entirely prevent the inference of any actionable information. The degree of this temporal dependence can be quantified by the so-called *autocorrelation* function [36].

Definition 1.5 (Lag-l sample autocorrelation)**.** Given a time series of length m with mean $\bar{v}$, its sample lag-l autocorrelation is

$$\rho_l = \frac{\sum_{t=l+1}^{m}(v_t - \bar{v})(v_{t-l} - \bar{v})}{\sum_{t=1}^{m}(v_t - \bar{v})^2}, \tag{1.3}$$

where $0 \leq l \leq m - 1$ [267, Equation 2.2].

A biomedical example of autocorrelated time series can be found in the measurement of blood sugar. We expect the blood sugar concentration at 7AM to be closer to the concentration measured at 8AM than to the one evaluated at 1PM (high lag-1 autocorrelation). While many methods for the detection and quantification of autocorrelation exist [18, 72, 194], they are beyond the scope of this thesis.

Assuming the time series of interest is weakly stationary, linear, and autocorrelated, we can set up one of the most simple parametric time series models, the autoregressive model.

Definition 1.6 (Autoregressive Model (AR(p))). The time series v_t is modelled as

$$v_t = \phi_0 + \phi_1 v_{t-1} + \cdots + \phi_p v_{t-p} + a_t, \tag{1.4}$$

where a_t refers to a white noise process as in Definition 1.4, ϕ to the parameters of the model, and $p \in \mathbb{N}_{>0}$ [267, Equation 2.9].

Note that we make use of the serial dependence/autocorrelation assumption by expressing the current observation v_t as a linear combination of its preceding values. An extension of the AR model where $p \to \infty$ results in another popular time series model, the moving-average model [267].

Definition 1.7 (Moving-Average Model (MA(q))). The time series v_t is modelled as

$$v_t = c_0 + a_t - \phi_1 a_{t-1} - \cdots - \phi_q a_{t-q}, \tag{1.5}$$

where c_0 is a constant, a_t refers to a white noise process as in Definition 1.4, and $q \in \mathbb{N}_{>0}$ [267, Equation 2.22].

A related field of methods for analysing time series is signal processing [188], which brought about popular techniques such as the Fourier transform [86] or the Wavelet-transform [190]. The expressive power of these approaches stems from representing and analysing signals in their frequency domain. The Fourier transform is a fundamental function transform that decomposes the input into its frequency *components*. Intuitively, it represents a signal as a weighted combination of sine and cosine waves of different frequencies. The set of resulting weights can be seen as a fingerprint of the signal in the frequency domain, which makes Fourier coefficients a powerful feature representation. A drawback of representing a time series by its frequency components is that all temporal information is lost. We know *which* frequencies are prevalent in a signal, but we do not know *when* they occur (i.e. frequency analysis is local in time). The so-called Wavelet-transform reintroduces the temporal component by performing multiple frequency analyses on *different scales*. The resulting scalogram is a faithful representation of the signal in both time- and frequency domain and has been used extensively in many biomedical applications [2, 83, 250]. While frequency-analyses such as the Wavelet-transform are informative representations, they do not align with the way a human may interpret and think about signals.

1.1.2 Subsequence View

A fundamental drawback of the "classical" models mentioned above is that most real-world time series are highly non-stationary (i.e. they exhibit a trend as shown in Figure 1.1) and,

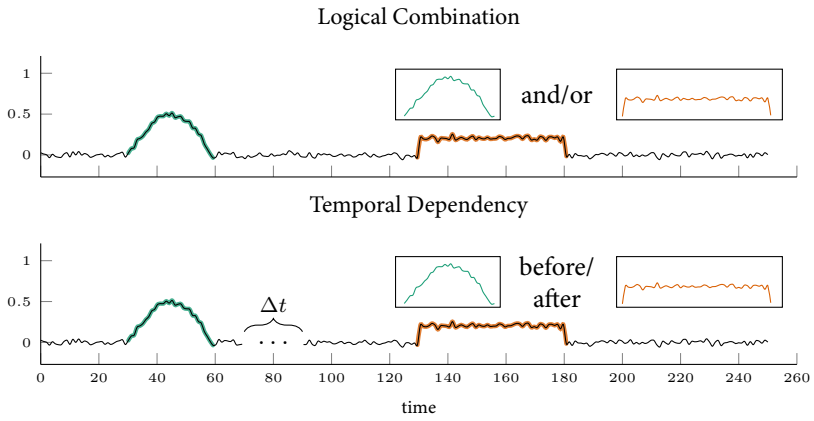


Figure 1.2: Subsequences as descriptive features of time series. The logical concatenation of two patterns might be indicative to which class a time series belongs. Similarly, the temporal order of pattern occurrences may indicate class membership.

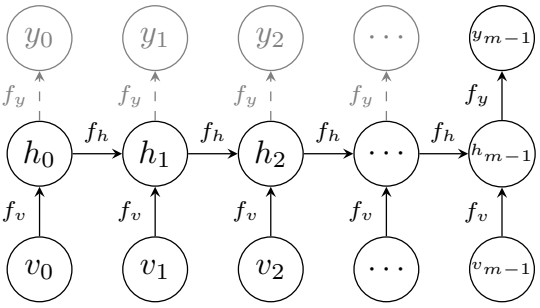
more importantly, identifying the model order (p or q) is non-trivial but a prerequisite for model fitting. Moreover, while extensions of these models exist (e.g. the ARMA(p, q) model or the ARIMA(p, d, q) model [36]) there is no guarantee that a given time series can be modelled by such approaches at all. In the specific case of biomedical data, it is also challenging to know what measurement features lead to a particular diagnosis. This requires an exploration of different time series features to determine which ones make comparisons and classifications meaningful.

In the context of this thesis, we are interested in flexible methods that make little assumptions about the nature of the time series and can learn interpretable and data set specific feature representations that are helpful for the task at hand (e.g. association mapping or classification). More specifically, in the chapters that follow, we will focus on analysing time series in terms of subsequences, dismissing any assumptions on the data-generating process. From a machine learning perspective, this allows us to derive interpretable and actionable time series features to discover temporal biomarkers (Chapter 2) and to classify time series of various kinds (Chapter 3, Section 3.2.3). Subsequences can be particularly meaningful features for time series in which complex patterns around specific values hold the majority of the information about the class label. Many vital parameters such as heart rate, blood pressure, or respiratory rate fall in this category of time series, where concrete values (e.g. systolic blood pressure above 120) define the health state of a patient. Subsequence-based methods can capture these complex patterns by explicitly taking this inductive bias into account. Figure 1.2 illustrates how using subsequences enables us to capture both logical and temporal dependencies, patterns that often characterise a specific subpopulation of interest. The upper

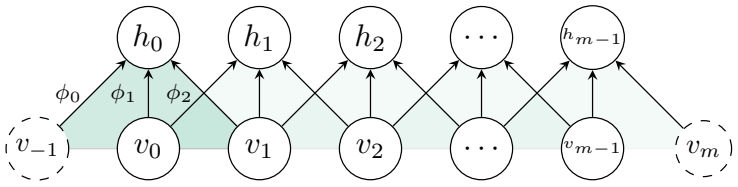
plot depicts classes of time series in which the combination or the individual appearance of a subsequence may be indicative of a class label. Below, we show that the "subsequence view" also allows us to define temporal dependencies between patterns over a certain time horizon Δt. A formal and comprehensive introduction of using subsequences as feature descriptors is provided in Section 2.1.

1.1.3 Time Series Representation Learning

Representing time series by their subsequences is advantageous as they are interpretable feature descriptors that guarantee (compared to the models from Section 1.1.1) to faithfully represent the input data. That being said, there are non-trivial modelling decisions to be made, including the subsequence selection procedure (e.g. subsequence length(s) and sliding window stride) and the precise feature computation, as we will see in Section 2.1. Moreover, subsequences are intrinsically inflexible representations as the data set predefines them. A priori, it is not clear whether they are the appropriate representation to solve the task at hand. By contrast, artificial neural networks (ANNs) (see Goodfellow et al. [93] are a successful machine learning concept that provides a flexible way of *learning* input representations that are task-dependent. ANNs for a thorough introduction) can be trained in ways that make them solve many real-world tasks with unprecedented effectiveness [228]. Two popular neural network-based approaches for time series analysis are illustrated in Figure 1.3. The first method, which takes the temporal order of sequential data explicitly into account, is the recurrent neural network (RNN) [108, 217], which is illustrated in Figure 1.3a. Temporally ordered "hidden" representations (h_t) are connected to their immediate neighbours, allowing them to utilise and propagate information from the past. The RNN is a flexible network architecture that allows generating multiple outputs (y_0 through y_{m-1}) that may be of interest in settings in which repeated predictions are required. In the context of time series classification (see Section 3.1 for an introduction), we would direct our focus on the final output y_{m-1} that aggregates information of the *complete* time series. An approach that was primarily developed for computer vision applications [139] is the convolutional neural network (CNN) [144]. Being rooted in the field of signal processing, the CNN's potential to analyse time series was recognised early on [143]. The main idea of CNNs is to distil local features by sliding a window (also called filter) over the input and learn a mapping into an intermediate representation. In the context of time series, it is crucial to mention that at each time point t, the filter aggregates information from past, present, and future. What makes these approaches particularly powerful is the fact that all parameters $\phi_\bullet$ in Figure 1.3 are *learnt* by optimising the generated output (e.g. by viewing y_{m-1} as predicted class label) using the backpropagation algorithm [217] and its variants [281]. As touched upon before,



(a) An "unrolled" illustration of a recurrent neural network (RNN). Each observation v_t is mapped into a hidden state h_t by a parameterised function of the form $f_\bullet = f(x, \phi_\bullet)$ for an input neuron $x \in \{v_0, \ldots, h_0, \ldots\}$. In the same way, f_h connects adjacent hidden states. Lastly, an output y_t can be learnt for each hidden state with a function f_y whose inputs are the learnable parameters ϕ_y and hidden state h.



(b) Illustration of a convolutional neural network (CNN). Filters (shown in green) slide over the input time series and aggregate multiple observations at once. In this example, the filter width is 3. The filter's parameters ϕ_1, ϕ_2, and ϕ_3 are *learnt* and weight individual observations. Dashed neurons imply that the input might require padding.

Figure 1.3: Two neural network paradigms to model time series data and learn flexible representations.

it is this training procedure that makes neural networks versatile learners of task-specific representations. The effectiveness of using neural networks in a biomedical context will be demonstrated in Section 3.3 by using a convolutional neural network to predict myocardial ischaemia from electrocardiograms (ECGs). Moreover, we will end this thesis by detailing a framework for analysing object-valued time series that benefit from neural networks' versatility.

1.1.4 Tasks & Challenges

General Time Series Tasks The most common, we might even say "classical" machine learning (ML) tasks in the analysis of real-valued time series include: 1. forecasting, i.e. pre-

dicting future values from historical ones [68, 78, 151], 2. anomaly detection, i.e. determining whether a new observation is normal or not [24], 3. change point detection, i.e. detecting (possibly transient) state changes in the data-generating process [7], 4. clustering, i.e. grouping time series of similar characteristics [3, 149], and 5. time series classification (TSC). However, the field of time series mining comprises additional tasks that evolve around the perspective illustrated in Figure 1.2, namely that time series are characterised by their subsequences. Before laying out the challenges that are addressed in this thesis, we will provide an overview of complementary time series mining tasks.

Time Series Mining Tasks As we will see in the chapter on TSC, obtaining a notion of (dis)similarity between two time series is the foundation of many classification algorithms. In addition, clustering methods or distance-based dimensionality reduction approaches such as multi-dimensional scaling [263] also necessitate the employment of an expressive distance measure. This renders the search for an appropriate time series distance measure an important *foundational* task in time series mining. The most commonly-used distance is the so-called Dynamic Time Warping (DTW) [220] distance, an alignment and dynamic programming-based approach. While being successfully used in many distance-based algorithms, it has the (theoretical and practical) drawback that it is not a metric in the mathematical sense, as it does not fulfil the triangle inequality [48]. However, there is also evidence that this non-metricity may be an advantage in practical applications [90]. More recent approaches include a differentiable version of DTW [61] or the matrix profile distance [91]. The identification of subsequences that are descriptive of a time series class (illustrated in Figure 1.2) turns our attention to another task: motif discovery. Motif discovery is the basis on which many downstream tasks such as clustering, rule discovery, and classification build upon. We therefore consider it to be the most important task in time series mining [5]. This notion is confirmed by a large body of literature [5, 67, 141, 173, 179, 236, 255, 265, 298, 299], and the fact that motifs lead to interpretable representations that convey semantically important local behaviour, a key property when working with medical time series.

Challenges Due to its general importance, motif discovery will play a central role in this thesis. We will focus on the intersection of motif discovery and time series classification and the challenges that arise when searching for motifs in a biomedical context. More specifically, we will develop solutions to the following challenges. **First**, while TSC approaches based on the extraction of motifs yield interpretable results and good predictive performance, the statistical association between detected motifs and class labels are neglected. This limits the descriptive power and interpretability of the results as the only statement we can make about

any detected subsequence pertains to its impact on classification accuracy. In the life sciences, however, we may be interested in finding temporal biomarkers that are *statistically associated* with a phenotype, a notion that goes beyond mere "predictability". In Chapter 2, we will therefore develop a motif discovery algorithm that is of inherently statistical nature. **Second**, many subsequence-based TSC algorithms only capture local characteristics of time series, preventing the algorithm to "get a full picture" of the data. Instead, it is common to use ensembles of algorithms, each of which with a different "view" of the data, to capture a time series comprehensively. Such ensemble methods, however, are computationally inefficient, as each model has to be trained and tuned individually. This will motivate us to develop a subsequence-based classification algorithm that considers local *and* global time series characteristics in Section 3.2. **The third** challenge we will address lies in the classification of complex phenotypes. In medicine, it can be expensive both in terms of cost and time, to obtain a reliable diagnosis. This is mainly due to the fact that the diagnostic process requires experience and elaborate examinations that may even put a burden on the patients. In the field of cardiology, the exercise stress test protocol is used to determine cardiovascular health in general, and ischaemic heart disease in particular. The practical utility of current screening techniques, however, is limited by either unfavourable diagnostic accuracy or by its obtrusive nature and high costs. In Section 3.3, we will investigate how we can use machine learning to increase diagnostic accuracy while reducing costs.

1.2 Object-Valued Time Series

Following the notion introduced in the beginning, an object-valued time series is composed of multiple "snapshots" of the same *structured* object that changes over time. The structured nature of the object determines the analysis methods we can use to investigate the object's trajectory in time. A prototypical example of an object-valued time series could be the sequence of a (Riemannian) manifold that is changing its intrinsic shape over time. Correspondingly, we may consider the object to move *along* a manifold. We therefore see this thesis as an "endemic part" of the field of manifold learning, which has started to provide an exciting perspective for biomedical research [8, 25, 176].

In the life sciences, object-valued time series are increasingly prevalent and frequently appear as sequences of *structured* static data, measured at different points in time. Biomedical examples include the cellular dynamics of single-cell RNA [261], time-varying functional magnetic resonance imaging (fMRI) data [209], dynamic protein-protein interaction networks [150], or dynamic electronic health record (EHR) graphs [146]. In general, time-varying graphs (biomedical or not) have recently gained tremendous attention from the ma-

chine learning community [239]. Being a universal data structure [33], this resurfacing interest in machine learning on graphs comes as no surprise and is subject of this thesis.

1.2.1 Tasks & Challenges

Many tasks from the analysis of real-valued time series are also important problems in object-valued time series. Examples of such tasks are anomaly detection in video data [241] or iterative graph "forecasting" as done by You et al. [291] for molecular graph generation. Despite the existence of these one-to-one correspondences, most tasks in the analysis of object-valued time series are specific to the object under investigation. One such object of particular interest to the machine learning community are artificial neural networks. Their empirical successes in many fields still surpass our theoretical understanding of their inner workings. One of the most fundamental challenges in deep learning is to understand the difference between neural networks that generalise well and those that do not [294]. While generalisation capabilities are measured in terms of out-of-sample error, the ability to generalise must be an inherent property of the network's configuration since once trained, the network no longer changes. Techniques such as network pruning [23] or applications of the lottery ticket hypothesis [87] support the notion that a trained neural network is composed of elements that are more important than others when it comes to generalising to unseen data. An essential question arising from this observation pertains to the possibility of deriving a formal measure that captures a neural network's generalisation capabilities by merely considering its structure. If such a measure exists, we expect it to be sensitive to common regularisation techniques that increase generalisation performance and it will allow us to develop an early stopping criterion without requiring a validation set. In the final chapter of this thesis, we present such a measure by viewing deep neural networks as time-varying graphs that can be investigated by means of persistent homology a technique from topological data analysis (TDA).

1.3 Contributions

In the first part of this section, we list and briefly summarise all contributions on which this thesis is based on. The second part of this section lists additional relevant scientific contributions by the author. If two or more authors equally contributed to a manuscript, their names are followed by a dagger symbol.

Real-Valued Time Series The computational and statistical problems of finding statistically validated temporal biomarkers are approached in the first contribution:

- **C. Bock**, T. Gumbsch, M. Moor, B. Rieck, D. Roqueiro, and K. Borgwardt. "Association mapping in biomedical time series via statistically significant shapelet mining". *Bioinformatics* 34:13, 2018, pp. i438–i446. DOI: 10.1093/bioinformatics/bty246.

In this paper, we develop Statistically Significant Shapelet Mining (S3M), a method to efficiently mine subsequences from univariate time series that are statistically associated with a binary class label. We build on the framework of significant pattern mining (SPM) to verify the statistical significance of such "shapelets" by utilising the idea of testability to reduce the number of hypotheses to correct for. Furthermore, we develop a contingency table pruning criterion to increase the algorithm's efficiency without sacrificing its effectiveness. To illustrate the merits of S3M, we apply it to three vital parameters from the MIMIC-III data set [124] identifying physiological signatures associated with sepsis. S3M eases the multiple testing burden when searching for temporal biomarkers, detects interpretable subsequences that are not only statistically validated but also good predictors of clinical endpoints in a classification setting.

Karsten Borgwardt proposed using Tarone's method to perform association mapping on time series subsequences. Christian Bock, Thomas Gumbsch, Michael Moor, Damian Roqueiro, and Karsten Borgwardt designed the study. Christian Bock and Thomas Gumbsch developed the prototype algorithm. Christian Bock, Bastian Rieck, and Michael Moor performed the experiments. Thomas Gumbsch developed the contingency table pruning procedure. Bastian Rieck contributed the proof and implemented an optimised variant of the algorithm. Christian Bock, Bastian Rieck, Thomas Gumbsch, and Karsten Borgwardt wrote the manuscript with contributions from all other authors.

Section 3.2 is based on the following contribution:

- **C. Bock**[†], M. Togninalli[†], E. Ghisu, T. Gumbsch, B. Rieck, and K. Borgwardt. "A Wasserstein Subsequence Kernel for Time Series". In: *2019 IEEE International Conference on Data Mining (ICDM)*. 2019, pp. 964–969. DOI: 10.1109/ICDM.2019.00108

In this work, we continue to leverage the expressiveness of time series subsequences to develop a novel time series classification method. We introduce the Wasserstein Time series Kernel (WTK), a method for the classification of univariate time series which utilises ideas from optimal transport (OT) theory and Reproducing Kernel Kreĭn Spaces (RKKS). The work is motivated by the observation that the straightforward application of Haussler's $\mathcal{R}$-convolution framework [104] to time series can become meaningless. When comparing two time series, WTK captures local *and* global characteristics resulting in a powerful representation leading to competitive classification accuracy across a wide variety of data sets.

For this work, the meaninglessness of the naïve application of the $\mathcal{R}$-convolution framework to subsequence-based time series kernels was brought forward by Karsten Borgwardt. Christian Bock, Bastian Rieck, Matteo Togninalli, and Karsten Borgwardt designed experiments and the study. Christian Bock, Elisabetta Ghisu, Thomas Gumbsch, Bastian Rieck, and Matteo Togninalli performed the experiments. All authors participated in the writing of the manuscript.

We conclude the time series classification chapter with a clinically motivated contribution developing a collaborative machine learning system for the classification of myocardial ischaemia:

- **C. Bock**, B. Rieck, J. Walter, I. Strebel, K. Borgwardt, and C. Müller. "Cardiologist-level prediction of stress-induced myocardial ischemia using multi-task learning." In preparation

We propose a deep learning approach for the classification of exercise-induced myocardial ischaemia (EIMI). EIMI is the pathophysiological hallmark of ischaemic heart or coronary artery disease (IHD/CAD), the leading cause of years of life lost (YLL) in Europe, the Americas, and Asia [120]. We leverage multi-task learning to predict this complex phenotype, whose determination is not only expensive but also burdensome for patients, by requiring only easy to obtain electrocardiograms and static patient data. Our system reaches cardiologist-level predictive performance decreasing the false positive rate while keeping high sensitivity. This way, we can lower the number of patients that undergo unnecessary radiation exposure. Incorporating the judgement of the treating physician increases predictive performance even more, leading to a reliable and collaborative decision support system for cardiologists.

Christian Bock, Bastian Rieck, Joan Walter, Karsten Borgwardt, and Christian Müller designed and evaluated the study. Ivo Strebel, Joan Walter, and Christian Müller collected the data. Ivo Strebel and Joan Walter prepared the data and data splits. Christian Bock preprocessed the data and performed the experiments with Bastian Rieck.

Object-Valued Time Series In the last chapter of this thesis, we go beyond real-valued time series and analyse time series of graph-structured data as done in

- B. Rieck[†], M. Togninalli[†], **C. Bock**[†], M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt. "Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology". In: *International Conference on Learning Representations (ICLR)*. 2019. DOI: 10.3929/ethz-b-000327207

More specifically, we propose neural persistence (NP), a complexity measure for artificial neural network architectures rooted in the field of topological data analysis (TDA). We view

feedforward neural networks as a collection of stratified graphs whose edge weights change during the training process. At each step of the training process, NP summarises the structural complexity of the network in a scalar value leading to a time series that describes the state of the network from a topological perspective. We show that neural persistence does not only reflect best practices developed in the deep learning community (e.g. dropout and batch normalisation) but can also be used as an effective early-stopping criterion that does not rely on a validation data set.

Christian Bock, Matteo Togninalli conveyed the original study idea, which Bastian Rieck refined and conceptualised. Christian Bock, Bastian Rieck, Matteo Togninalli, Michael Moor, and Max Horn designed the study. Bastian Rieck contributed theorems and their proofs together with Michael Moor. Christian Bock, Bastian Rieck, Matteo Togninalli, Michael Moor, Max Horn, and Thomas Gumbsch performed the experiments. All authors contributed to the writing of the manuscript.

In addition to the works listed above, the author also contributed to the following publications:

- B. Rieck†, **C. Bock**†, and K. Borgwardt. "A Persistent Weisfeiler-Lehman Procedure for Graph Classification". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5448–5458. URL: https://proceedings.mlr.press/v97/rieck19a.html

- B. Rieck†, T. Yates†, **C. Bock**, K. Borgwardt, G. Wolf, N. Turk-Browne, and S. Krishnaswamy. "Uncovering the Topology of Time-Varying fMRI Data using Cubical Persistence". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6900–6912. arXiv: 2006.07882 [q-bio.NC]

- M. Horn, M. Moor, **C. Bock**, B. Rieck, and K. Borgwardt. "Set Functions for Time Series". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4353–4363. arXiv: 1909.12064 [cs.LG]

- S. L. Hyland†, M. Faltys†, M. Hüser†, X. Lyu†, T. Gumbsch†, C. Esteban, **C. Bock**, M. Horn, M. Moor, B. Rieck, et al. "Early prediction of circulatory failure in the intensive care unit using machine learning". *Nature medicine* 26:3, 2020, pp. 364–373. DOI: 10.1038/s41591-020-0789-4

- T. Gumbsch, **C. Bock**, M. Moor, B. Rieck, and K. Borgwardt. "Enhancing statistical power in temporal biomarker discovery through representative shapelet mining". *Bioinformatics* 36:Supplement_2, 2020, pp. i840–i848. DOI: 10.1093/bioinformatics/btaa815

- **C. Bock**[†], M. Moor[†], C. R. Jutzeler, and K. Borgwardt. "Machine learning for biomedical time series classification: from shapelets to deep learning". In: *Artificial Neural Networks*. Springer, 2021, pp. 33–71. DOI: 10.1007/978-1-0716-0826-5_2

Part I

Real-Valued Time Series

2 Pattern Mining for Time Series

> In which we extend the significant pattern mining framework to time series and extract sepsis-associated subsequences from physiological signals of ICU patients.

We begin the analysis of real-valued time series by framing the search for informative time series motifs as a significant pattern mining (SPM) problem [157]. First, we present the idea of "shapelets" [288], the foundation for this and the following chapter in Section 2.1. Then, a brief, yet self-contained introduction into the field of significant pattern mining is provided. This includes a description of the general problem that SPM is concerned with, and is followed by laying out one of the most relevant problems in SPM: the multiple hypothesis testing problem. We conclude Section 2.2 by introducing the concepts of minimum p-value and testability. We will then present a new method that extends the SPM framework to univariate time series data in Section 2.3. This chapter is based on the following publication:

- **C. Bock**, T. Gumbsch, M. Moor, B. Rieck, D. Roqueiro, and K. Borgwardt. "Association mapping in biomedical time series via statistically significant shapelet mining". *Bioinformatics* 34:13, 2018, pp. i438–i446. DOI: 10.1093/bioinformatics/bty246

2.1 Shapelets

Shapelets, first introduced by Ye and Keogh [288], are short time series subsequences developed to maximise predictive power in time series classification tasks. Due to their interpretability and good classification performance, shapelet-based classifiers were developed for a wide range of medical (and non-medical applications) [89, 94, 191, 254, 285]. Figure 2.1 visualises the main idea of a shapelet: The red subsequence in the centre appears in all three time series from the positive class ($y = 1$), and never in the negative class ($y = 0$). This subsequence is therefore *characteristic* for the positive class and an algorithm that detects it in a time series will classify the time series as belonging to class 1. We refer to a subsequence that maximises the prediction performance of such a classifier as *shapelet*. Constructing a

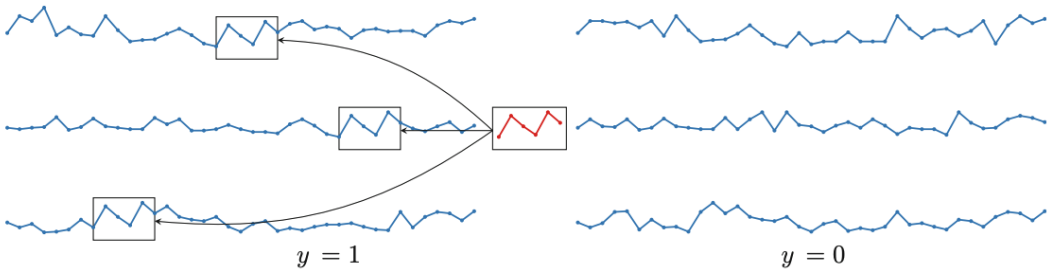


Figure 2.1: A shapelet (red) is a time series motif or pattern that is enriched in certain class. Here, the shapelet appears in all time series belonging to class $y = 1$. Reproduced from [26] with permission from Oxford University Press.

shapelet-based classifier therefore requires 1. a collection of candidate shapelets $\mathcal{M}$, 2. a pattern indicator function g that determines whether a given subsequence *occurs* in a time series, and 3. a measure of classification performance. In the following sections, we will introduce the fundamental concepts of shapelet mining, starting with the extraction of shapelet candidates, followed by a notion of distance between shapelets and time series and the construction of a simple shapelet-based classifier.

2.1.1 Notation & Shapelet Candidate Extraction

Before we embark on the journey of finding the most characteristic subsequences for a given class label, it is necessary to define a set of subsequences that we want to investigate. For the remainder of this section, we will restrict $\mathcal{M}$ to originate from the data set under investigation (in contrast to generating random length-w sequences). Let us denote the univariate time series T of length m as the sequence of $m > 0$ tuples $T = ((t_1, v_1), \ldots, (t_m, v_m))$. In other words, each observation is represented by its observed value $v_i \in \mathbb{R}$ and respective observation time $t_i \in \mathbb{R}$ (or $t_i \in \mathbb{N}$). We denote a time series data set with k samples and binary class label $y \in \{0, 1\}$ as the set of tuples $\mathcal{T} = \{(T_1, y_1), \ldots, (T_k, y_k)\}$. Furthermore, we denote a length-w subsequence extracted from the k^{th} time series beginning at time point t_b as the sequence of its observed values $S^{[k]}_{b,w} = (v^{[k]}_b, \ldots, v^{[k]}_{b+w-1})$.

To extract shapelet candidates, we will use a sliding window approach with window size w and stride $s = 1$. This means that, starting from t_1, we extract the set of w-length subsequences $\mathcal{M}_w$ as defined below

$$\mathcal{M}_w := \{S^{[j]}_{1,w}, S^{[j]}_{1+s,w}, \ldots, S^{[j]}_{m^{[j]}-w+1,w}\}^k_{j=1}, \tag{2.1}$$

where $m^{[k]}$ denotes the length of time series k. Note, to prevent the extraction of largely overlapping subsequences, it can be sensible to choose $s > 1$, however, to maximise the search space, we will enumerate *all* length-w subsequences. In general, this leads to

$$\lceil (\frac{m^{[k]} - w + 1}{s} \rceil \tag{2.2}$$

subsequences per time series. A complete set of shapelet candidates may contain subsequences of multiple lengths w_1, …, w_u and will be constructed by the union $\mathcal{M} = \bigcup_{i=1}^{u} \mathcal{M}_{w_i}$.

2.1.2 Subsequence Pseudo-Distance

After establishing a set of candidate shapelets, a pattern indicator function must be defined to determine in a binary fashion whether a candidate $\zeta \in \mathcal{M}$ occurs in time series $T^{[k]}$. A simple approach is to apply an "exact match" requirement:

$$g_\zeta(T^{[k]}) = \begin{cases} 1, & \text{if there exists a } b \in \mathbb{N}_1 \text{ s.t. } S^{[k]}_{b,|\zeta|} = \zeta \\ 0, & \text{else} \end{cases} \tag{2.3}$$

However, this is an extremely strict criterion as it does not allow for the smallest deviation. On real world data sets, however, the general shape of a subsequence (e.g. the increase, decrease, or fluctuation of observations) are more interesting than *exact* matches. In order to derive a more lenient pattern indicator function, we define the function $d(\zeta, T)$ to be the smallest Euclidean distance between ζ and all length-w subsequences of T. If $d(\zeta, T)$ is smaller than a threshold θ, we say ζ appears in T. Formally, we have the Euclidean distance between two *equally long* sequences $d_{\text{Euc}}(S^{[a]}, S^{[b]}) = \sqrt{\sum_{i=1}^{m} \left(v_i^{[a]} - v_i^{[b]} \right)^2}$, from which we define the following pseudo-distance:

Definition 2.1 (Subsequence Pseudo-Distance)**.** Given a length-w shapelet candidate ζ and a longer time series T, their distance is

$$d(\zeta, T) = \min_j d_{\text{Euc}}(\zeta, S_{j,w}). \tag{2.4}$$

Now, a threshold-based pattern indicator function can be written as

$$g_\zeta(T^{[k]}, \theta) = \begin{cases} 1, & \text{if } d(\zeta, T) < \theta \\ 0, & \text{else} \end{cases} \tag{2.5}$$

The *pattern* is no longer a single shapelet candidate but the *combination* of subsequence and threshold θ.

It is of interest to note that $d(\cdot,\cdot)$ is not a metric in the mathematical sense (hence, *pseudo*-distance), as it is only defined if the cardinality of the second argument is greater then or equal to the cardinality of the first argument. Per construction of ζ (w will be much smaller than the length of most time series in the data set), the equality $|\zeta| = |T|$ will almost never hold, which prevents us from even *checking* the symmetry ($d(a,b) = d(b,a)$) and triangle inequality property ($d(a,c) \leq d(a,b) + d(b,c)$). Furthermore, we can show that the identity of indiscernibles ($d(a,b) = 0 \Leftrightarrow a = b$) is not guaranteed, as in most cases ζ and T are not even of the same cardinality (and we did not define a notion of identity for such sequences).

2.1.3 A Simple, Shapelet-Based Classifier

To conclude the introduction of shapelets, we will now detail how Ye and Keogh [288] put them to use in a classification context using entropy-based decision rules. The pattern indicator function in Equation 2.5 allows us to split $\mathcal{T}$ into two sets: One which contains all time series whose distance to ζ is less than θ ($\mathcal{T}_{\zeta<\theta}$), and one with the remaining time series ($\mathcal{T}_{\zeta\geq\theta}$). The quality of this split can be measured in terms of *information gain* (or mutual information), i.e. the difference between the entropy of the data set before and after the split. It measures the reduction in uncertainty about the class label resulting from learning about the occurrence of the pattern [165]. Let $P_{\mathcal{T}}(y)$ be the fraction of samples in $\mathcal{T}$ whose class label is y, then the class label's (binary) *entropy* is

$$H_{\mathcal{T}}(Y) = -P_{\mathcal{T}}(y_0)\mathrm{ld}P_{\mathcal{T}}(y_0) - P_{\mathcal{T}}(y_1)\mathrm{ld}P_{\mathcal{T}}(y_1), \tag{2.6}$$

where ld refers to the logarithmus dualis. From this we see, that $P_{\mathcal{T}}(y)$ reaches its maximum of 1 if $P_{\mathcal{T}}(y_0) = P_{\mathcal{T}}(y_1) = \frac{1}{2}$ and its minimum of 0 if the data set consists of samples from one class only.

When we apply the aforementioned split on the data set, the *conditional entropy* measures the average uncertainty about the class label that remains when we know about the pattern's occurrence:

$$H_{\mathcal{T}}(Y|\zeta,\theta) = \frac{|\mathcal{T}_{\zeta<\theta}|}{|\mathcal{T}|} H_{\mathcal{T}_{\zeta<\theta}}(Y) + \frac{|\mathcal{T}_{\zeta\geq\theta}|}{|\mathcal{T}|} H_{\mathcal{T}_{\zeta\geq\theta}}(Y). \tag{2.7}$$

In other words, it is the sum of two entropy values weighted by the relative size of each split. Now, we can measure how well a given subsequence/threshold pair splits a data set in terms of its information gain:

$$I_{\text{gain}}(\mathcal{T}, \zeta, \theta) = \overbrace{H_{\mathcal{T}}(Y)}^{\text{Entropy before split}} - \overbrace{H_{\mathcal{T}}(Y|\zeta, \theta)}^{\text{Entropy after split}} . \tag{2.8}$$

With a well-defined performance measure at hand, we can now formulate the search for the best shapelet (i.e. subsequence/treshold pair) as the following optimisation problem:

$$(\zeta_{\text{best}}, \theta_{\text{best}}) = \underset{\zeta \in \mathcal{M}, \theta \in \Theta}{\operatorname{argmax}} \; I_{\text{gain}}(\mathcal{T}, \zeta, \theta) \tag{2.9}$$

Note that up to this point, we did not introduce a procedure to derive the threshold set Θ. This was a deliberate choice and we will detail the standard approach when introducing our method in Section 2.3. Lastly, we should mention that Equation 2.9 yields a very simple "classifier" of limited expressivity as it consists of only *one* shapelet. In their original work, Ye and Keogh [288] proposed a decision tree classifier incorporating multiple shapelets, and by now, shapelet-based methods are considered a distinct class of algorithms for times series classification [12].

While shapelet-based approaches have the advantage of being interpretable in the sense that they are actual subsequences from the data set, they are not statistically validated. In this context, statistical validation refers to the computation of a p-value quantifying the association of a shapelet with a binary class label. In biomedicine and the life sciences, where the class label can indicate the presence/absence of a phenotype of interest, such statistical tests yield a notion of interpretability that clinicians are familiar with. To mine temporal biomarkers that exhibit both levels of interpretability (due to their shape and due to their statistical association), we must combine the shapelet mining procedure with statistical testing. To do so, we will make use of the significant pattern mining framework as introduced in the following section.

2.2 Significant Pattern Mining

Significant pattern mining is a branch of machine learning that tackles the computational and statistical challenges arising from searching for statistical associations of interacting features with a binary class label. Applied to biomedical time series, SPM can help identify subsequences of physiological measurements that are statistically associated with a phenotype of interest (e.g. sepsis). Until recently, SPM algorithms were predominantly used for the

analysis of static feature datasets such as *itemsets* and *graph-structured data*. In the following sections we lay out the conceptual and mathematical foundations for a novel SPM approach for *time series data* which we will develop in Section 2.3. While this introduction is complete and self-contained, we refer to Llinares-López [157] and Gumpinger [98], who provide a thorough presentation of the field of SPM.

2.2.1 Problem Statement

Consider the dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where each of n independent and identically distributed (i.i.d.) samples is represented by a feature indicator set x and a binary class label $y \in \{0, 1\}$. The set $\mathcal{J} = \{1, \ldots, d\}$ indexes all d features that may exist in a given data set. Each sample can contain its own subset of features: $x_i \subseteq \mathcal{J}$. For example, the feature set $x_i = \{2, 3, 5\}$ indicates that sample i exhibits features 2, 3, and 5. A pattern $\mathcal{S}$ is any discrete substructure of $\mathcal{J}$ ($\mathcal{S} \subseteq \mathcal{J}$). To indicate the presence of a pattern in a sample, we introduce the *pattern indicator function* $g_s(x)$ as follows:

Definition 2.2 (Pattern indicator function)**.** Given a pattern $\mathcal{S}$ and the feature vector of the i^{th} sample x_i, we have the indicator function

$$g_{\mathcal{S}}(x_i) = \begin{cases} 1, & \text{if } \mathcal{S} \subseteq x_i \\ 0, & \text{else} \end{cases}. \tag{2.10}$$

As mentioned before, SPM aims to find patterns (or higher-order features interactions) for which a statistically significant association with the binary class label y exists. We will use the pattern indicator function for constructing a *contingency table*, which can be analysed using (discrete) test statistics such as the Fisher's exact test [84] or the χ^2 test [193]. These statistical tests allow us to quantify the association of $\mathcal{S}$ and y in terms of a p-value.

2.2.2 Hypothesis Testing

The goal of statistical hypothesis testing is to shed light on the (in)dependence of two or more random variables. In our case, we model the binary class indicator and the pattern indicator as random variables Y and $G_{\mathcal{S}}$. Informally, we want to know if information about the existence of pattern $\mathcal{S}$ in sample i entails any information about its class label y_i. If this was not the case, the joint probability distribution factorises into the product $P(Y, G_{\mathcal{S}}) = P(Y)P(G_{\mathcal{S}})$. For the time being, we hypothesise that this factorisation holds until we are convinced otherwise. We call this hypothesis (i.e. pattern and class label are independent) the *null hypothesis* (H_0).

In any machine learning problem, we only have access to a finite number of samples and the true, data-generating probability distributions are unknown. Hence, it is not possible to assess their independence. Taking a frequentist view, however, allows us to approximate the joint distribution of Y and $G_\mathcal{S}$ using the counts of the following four events:

1. $(y_i = 1, g_\mathcal{S}(x_i) = 1)$
2. $(y_i = 1, g_\mathcal{S}(x_i) = 0)$
3. $(y_i = 0, g_\mathcal{S}(x_i) = 1)$
4. $(y_i = 0, g_\mathcal{S}(x_i) = 0)$,

where $i = \{1, \ldots, n\}$. A more compact representation of these counts is the 2×2 contingency table as shown in Table 2.1.

Table 2.1: Contingency table with counts of four events in a data set of size n.

	$g_\mathcal{S} = 1$	$g_\mathcal{S} = 0$	
$y = 1$	$a_\mathcal{S}$	$b_\mathcal{S}$	$n_1 = a_\mathcal{S} + b_\mathcal{S}$
$y = 0$	$c_\mathcal{S}$	$d_\mathcal{S}$	$n_0 = c_\mathcal{S} + d_\mathcal{S}$
	$r_\mathcal{S}$	$q_\mathcal{S}$	$n = n_1 + n_0 = r_\mathcal{S} + q_\mathcal{S}$

The cells $a_\mathcal{S}$ through $d_\mathcal{S}$ can be computed from the data set as follows:

$$a_\mathcal{S} = \sum_{i=1}^{n} g_\mathcal{S}(x_i) y_i \qquad b_\mathcal{S} = \sum_{i=1}^{n} (1 - g_\mathcal{S}(x_i)) y_i$$
$$c_\mathcal{S} = \sum_{i=1}^{n} g_\mathcal{S}(x_i)(1 - y_i) \qquad d_\mathcal{S} = \sum_{i=1}^{n} (1 - g_\mathcal{S}(x_i))(1 - y_i).$$

The core idea of association tests, such as Fisher's exact test [84] or Pearson's χ^2 test [193] , is to derive a test statistic $T\colon \{(x_i, y_i)\}_{i=1}^n \to \mathbb{R}$ and establish its *null distribution*.

Definition 2.3 (Null Distribution)**.** The distribution of the scalar test statistic under the assumption that the null hypothesis is true: $P(T = t|H_0)$.

In both tests, T is derived from the contingency table of the data while modelling $a_\mathcal{S}$ as a realisation of a random variable $A_\mathcal{S}$. An informative test statistic is chosen in a way that $T(a_\mathcal{S})$ of a data set with associated Y and $G_\mathcal{S}$ is sufficiently different from $T(a_\mathcal{S})$ of a data set with independent Y and $G_\mathcal{S}$. If this gap is large enough, we claim we are convinced that the hypothesis of independence does not hold. In the following paragraphs we will detail Pearson's χ^2 test as it is the procedure on which our method in Section 2.3 will rely on.

2.2.2.1 Pearson's χ^2 Test

The test statistic in Pearson's χ^2 test builds upon the insight that with fixed marginal counts $r_\mathcal{S}$, n, and n_1, $a_\mathcal{S}$ follows a hypergeometric distribution [157]

$$\begin{aligned} P(a_\mathcal{S}|n_1, r_\mathcal{S}, H_0) &= \text{hypergeom}(a_\mathcal{S}|n, n_1, r_\mathcal{S}) \\ &= \frac{\binom{n_1}{a_\mathcal{S}}\binom{n-n_1}{r_\mathcal{S}-a_\mathcal{S}}}{\binom{n}{r_\mathcal{S}}} \end{aligned} \tag{2.11}$$

with mean and variance [276]

$$\mathbb{E}[a_\mathcal{S}|n, n_1, r_\mathcal{S}] = \frac{r_\mathcal{S} n_1}{n} \tag{2.12}$$

$$\text{Var}(a_\mathcal{S}|n, n_1, r_\mathcal{S}) = n_1 \frac{r_\mathcal{S}}{n}\left(1 - \frac{r_\mathcal{S}}{n}\right)\left(\frac{n - n_1}{n - 1}\right) \tag{2.13}$$

$$= n_1 \frac{r_\mathcal{S}(n - r_\mathcal{S})(n - n_1)}{n^2(n - 1)}. \tag{2.14}$$

Both moments can be used to derive the t-statistic by converting $a_\mathcal{S}$ into a scalar Z:

$$Z = \frac{a_\mathcal{S} - \mathbb{E}[a_\mathcal{S}|n, n_1, r_\mathcal{S}]}{\sqrt{\text{Var}(a_\mathcal{S}|n, n_1, r_\mathcal{S})}} \tag{2.15}$$

$$= \frac{a_\mathcal{S} - \frac{r_\mathcal{S} n_1}{n}}{\sqrt{n_1 \frac{r_\mathcal{S}(n-r_\mathcal{S})(n-n_1)}{n^2(n-1)}}}. \tag{2.16}$$

This transformation is similar to the process of score standardisation. Under the null hypothesis, this quantity converges to a standard normal distribution (i.e. a normal distribution with zero mean and unit variance) $\mathcal{N}(0, 1)$ if the data are i.i.d. and n is large (central limit theorem). It will be small when the observed pattern count $a_\mathcal{S}$ aligns with the null hypothesis (i.e. $a_\mathcal{S}$ is close to what we would expect given n, n_1, and $r_\mathcal{S}$) and large when expectation and observation diverge. Due to the convergence of Z to $\mathcal{N}(0, 1)$, its square Z^2 follows a χ^2 distribution with one degree of freedom (a χ^2_k distribution has k degrees of freedom and represents the distribution of the sum of squares of k independent random variables each of which follow a standard normal distribution).

Equipped with a test statistic, the p-value of Pearson's χ^2 test can be computed using the cumulative density function (CDF) of the χ^2_1-distribution $F_{\chi^2_1}$:

$$p_\mathcal{S}(a_\mathcal{S}|n, n_1, r_\mathcal{S}) = 1 - F_{\chi^2_1}(a_\mathcal{S}|n, n_1, r_\mathcal{S}). \tag{2.17}$$

The p-value represents the probability of obtaining a test statistic at least as extreme as the one we observe, assuming the pattern and class label are independent (null hypothesis). If this probability is low enough (i.e lower than a pre-defined significance level α), we reject the independence assumption. When rejecting independence, we commonly infer a dependency between pattern and label. It is important, however, to stress that the convergence of Z to a standard normal depends on a large sample size n and the fact that all samples are identically and independently distributed. When dealing with small data sets or data that violate the i.i.d. condition, the χ^2-test will yield less reliable results.

2.2.2.2 Multiple Hypothesis Testing

In the introduction of this chapter, we set out to find statistical associations between class membership and higher-order *feature interactions* (i.e. feature combinations). At this point, we only know how to assess the statistical association of a *single* pattern: 1. We define a test statistic, 2. we derive a null distribution, 3. we compute a p-value, and 4. we reject the null hypothesis, or we do not. If we model the p-value as a random variable X with F as its CDF, we can show that under the null hypothesis, X is uniformly distributed, which means $P(F(X) \leq f) = f$. The uniform distribution of p-values under H_0 is crucial to define a significance threshold α that is equivalent to the probability of rejecting a true null hypothesis (making a type I error). Rejecting a hypothesis whose p-value is $p \leq \alpha$ is equivalent to setting α as an upper limit on how "likely" we allow the hypothesis to be in order to be rejected. *Only if* p-values follow a uniform distribution under the null hypothesis, can α be used to control the false positive (FP) rate (i.e. the number of falsely rejected true null hypotheses) this way. As an example, if we set $\alpha = 0.05$, as done in many applications, we accept a 5 % chance that we falsely reject a true null hypothesis.

When we test multiple hypotheses simultaneously, and correct each of them at the same significance threshold, the chance of a false positive grows beyond α. This scenario is referred to as the *multiple hypothesis testing problem* in which the sheer amount of tested hypotheses (think "for each pattern a hypothesis test"), leads to a large number of incorrect associations. The probability of making *at least one* such wrong association (given a significant threshold δ) is also called family-wise error rate (FWER) and formalised as

$$\text{FWER}(\delta) := P(\text{number of FPs } \geq 1 | \delta). \tag{2.18}$$

To counteract the multiple hypothesis problem, we need to adjust the significance threshold we apply to each *individual* hypothesis, such that the *overall* FWER meets α. Formally, we are interested in finding a corrected threshold $\delta \leq \alpha$ such that $\text{FWER}(\delta) \leq \alpha$. An ap-

proach commonly employed in practice is Bonferroni's correction procedure [31]. Let $\mathcal{H}$ be the set of all hypotheses to test, then the Bonferroni corrected significance threshold is

$$\delta_{\text{Bon}} = \alpha/|\mathcal{H}|. \tag{2.19}$$

It can be shown (e.g. by Gumpinger [98]) that this correction *guarantees* the control of FWER but can be *extremely* conservative when correcting for thousands or millions of hypotheses. As mentioned before, in SPM we are concerned with higher-order feature interactions which can lead to a massive amount of pattern combinations for which a hypothesis test is needed. A correction factor that is too stringent can lead to a high number of "missed" true associations (i.e. most p-values exceed δ_{Bon}) and therefore to a loss of statistical power. To find a balance between such type II errors and the number of type I errors, it is imperative to find less conservative correction procedures.

2.2.3 Minimum p-value and Testability

The corrected threshold we are interested in maximises statistical power while controlling the FWER. We denote this "ideal" threshold as δ^* and embark on a journey to find a δ for which $\delta_{\text{Bon}} \ll \delta \leq \delta^*$. Such a threshold can be found following Tarone's approach [257], which we will detail in this section.

Tarone's procedure builds upon the idea of *testable hypotheses* and the insight that *non-testable hypotheses* can never become significant and will therefore not contribute to the FWER. This, in turn, implies that a correction for non-testable hypotheses is not necessary, as detailed below.

Minimal Attainable p-value Recall the contingency table from Section 2.2.2, where n is the data set size, n_1 the number of positively labelled samples therein, and $r_{\mathcal{S}}$ the number of *all* samples that contain the pattern $\mathcal{S}$. Together with $a_{\mathcal{S}}$ (the number of positively labelled samples that contain $\mathcal{S}$), these variables uniquely define the contingency table. Additionally, for any given $\mathcal{S}$, $r_{\mathcal{S}}$ will be fixed and the only degree of freedom lies in $a_{\mathcal{S}}$. In other words, we can get a finite number of table configurations with the same table margins, n, n_1, and $r_{\mathcal{S}}$, by varying $a_{\mathcal{S}}$. We can see how the margins bound the values of $a_{\mathcal{S}}$ by considering the extremal cases: Either the pattern is observed in all samples from the negative class ($a_{\mathcal{S}}^{\text{min}}$), or in all samples from the positive class ($a_{\mathcal{S}}^{\text{max}}$):

$$a_{\mathcal{S}}^{\text{min}} := \max\left(n_1 + r_{\mathcal{S}} - n, 0\right) \tag{2.20}$$

$$a_{\mathcal{S}}^{\text{max}} := \min\left(r_{\mathcal{S}}, n_1\right) \tag{2.21}$$

The set of all possible values for $a_{\mathcal{S}}$ can then be expressed as as all natural numbers in these bounds:

$$\mathcal{A}_{\mathcal{S}} := \{a_{\mathcal{S}} | a_{\mathcal{S}}^{\min} \leq a_{\mathcal{S}} \leq a_{\mathcal{S}}^{\max}\}, \text{ with } a_{\mathcal{S}} \in \mathbb{N}. \tag{2.22}$$

Finding the smallest p-value among these table configurations (see Equation 2.23) is equivalent to finding the pattern that exhibits the strongest association with the class label:

$$p_{\mathcal{S}}^{\min} := \min \{p_{\mathcal{S}}(a_{\mathcal{S}} | n, n_1, r_{\mathcal{S}}) | a_{\mathcal{S}} \in \mathcal{A}_{\mathcal{S}}\} \tag{2.23}$$

Luckily, it is not necessary to iterate over all table configuration and evaluate their p-values. In fact, there is a closed-form solution to compute the minimal attainable p-value for Pearson's χ^2 test [157] which is shown in Equation 2.24.

$$p_{\mathcal{S}}^{\min}(r_{\mathcal{S}}) := \begin{cases} 1 - F_{\chi_1^2}\left((n-1)\frac{n_b}{n_a}\frac{r_{\mathcal{S}}}{n-r_{\mathcal{S}}}\right) & \text{if } 0 \leq t_{\mathcal{S}} \leq n_a \\ 1 - F_{\chi_1^2}\left((n-1)\frac{n_a}{n_b}\frac{n-r_{\mathcal{S}}}{r_{\mathcal{S}}}\right) & \text{if } n_a \leq r_{\mathcal{S}} \leq \frac{n}{2} \\ 1 - F_{\chi_1^2}\left((n-1)\frac{n_a}{n_b}\frac{r_{\mathcal{S}}}{n-r_{\mathcal{S}}}\right) & \text{if } \frac{n}{2} \leq r_{\mathcal{S}} \leq n_b \\ 1 - F_{\chi_1^2}\left((n-1)\frac{n_b}{n_a}\frac{n-r_{\mathcal{S}}}{r_{\mathcal{S}}}\right) & \text{if } n_b \leq r_{\mathcal{S}} \leq n \end{cases}, \tag{2.24}$$

where $n_a := \min(n_1, n - n_1)$ and $n_b := \max(n_1, n - n_1)$. Tarone [257] used the notion of minimum p-value to derive a less stringent significant threshold δ_{Tar}, which is defined as follows: Let $\mathcal{H}_{\text{testable}}(\delta)$ be the set of all patterns whose minimum p-value is less than a predefined threshold δ, then $\delta_{\text{Tar}} = \alpha / |\mathcal{H}_{\text{testable}}(\delta)|$, with

$$\mathcal{H}_{\text{testable}}(\delta) = \{\mathcal{S} | p_{\mathcal{S}}^{\min}(r_{\mathcal{S}}) \leq \delta\}. \tag{2.25}$$

In the context of SPM, Tarone's method has been investigated and used in multiple applications such as graph mining [98, 252] or gene regulation [259]. In Section 2.3, we will use and extend Tarone's method to enable efficient mining of temporal patterns in time series data.

2.3 Statistically Significant Subsequence Mining (S3M)

In this section, we describe a novel method that combines the ideas from SPM and shapelets to discover temporal patterns in time series data. Throughout the description of our method, we will abbreviate "statistically significant" with the term "significant".

Let $\mathcal{W}$ be the set of all integer-valued subsequence lengths of interest, i.e. $\mathcal{W} := \{w_{\min}, w_{\min+1}, \ldots, w_{\max}\}$. Moreover, note that the number of w-length subsequences was given in Equation 2.2 on page 21. As we set the stride of the window to $s = 1$, we

perform an exhaustive search over all possible candidates according to $\mathcal{W}$[1]. Furthermore, $\mathcal{D} := (d_0, \ldots, d_{k-1})$ is the ordered sequence of distances (following Definition 2.1) between shapelet candidate $\mathcal{S} \in \mathcal{M}$ and k time series. In the classification setting as introduced by Ye and Keogh [288], we use distances of the form $(d_i+d_{i+1})/2$ as threshold θ for indicator function $g_\zeta(\cdot, \theta)$ (see Equation 2.5). This way, each threshold maximises the separation margin between the two classes, which is necessary to optimise predictive performance on unseen data. We also consider a threshold slightly lower than d_0 and slightly higher than $d_{k=1}$. This leads to $k+1$ thresholds per shapelet candidate (all "midway" distances and a leading and trailing threshold on both ends of the list)[2]. Combining this with the number of candidates each $w \in \mathcal{W}$ "generates", the naïve Bonferroni-corrected significance threshold for a data set with k time series of length m is:

$$\delta_{\text{Bon}} = \frac{\alpha}{k \underbrace{(k+1)}_{\text{no. thresholds}} \underbrace{\left(\sum_{w=w_{\min}}^{w_{\max}} (m-w+1) \right)}_{\text{no. candidates}}} \tag{2.26}$$

This means, that even for a small data set containing 100 time series of length 100, $w_{\min} = 5$, and $w_{\max} = 10$, the correction factor will be in the order of 5×10^6, which will lead to a significant loss of statistical power. Similarly, the exhaustive extraction of all subsequences of length $w = 1, \ldots, m$ results in a space complexity of

$$\mathcal{O}\big((k+1)km^2\big), \tag{2.27}$$

where km^2 refers to the total number of subsequences to be held in memory, for each of which $k+1$ contingency tables are required. The factor m^2 in Equation 2.27 comes from the fact that

$$\sum_{w=1}^{m} (m-w+1) = \frac{1}{2} m(m+1).$$

Possible solutions to the high memory footprint are discussed at the end of this chapter.

To remedy the problem of reduced statistical power, we will now describe an iterative pruning algorithm that makes the usage of Tarone's method [257] (as described in Section 2.2.3) more efficient and feasible for the application in medium-sized data sets. A detail about

[1] The size of the search space constitutes the computational bottleneck of our method which is further discussed in Section 2.5.

[2] For the construction of contingency tables, there is no need to use midway distances since it is not our objective to find thresholds that maximise predictive performance. In the association testing framework, the number of thresholds in Equation 2.26 is therefore reduced by 1.

Tarone's adjustment procedure we did not mention yet, is its iterative character requiring continuous updates of the testability threshold to guarantee control of the FWER: We start with a significance threshold $\hat{\delta} := \alpha$, where α is the target FWER. Then, each shapelet/threshold pair $\mathcal{S} = (\zeta, \theta)$ will be processed individually. If its minimum p-value $p_{\mathcal{S}}^{\min} \leq \hat{\delta}$, the pattern is added to a list of testable patterns $\mathcal{H}_{\text{testable}}$. As this addition will change the the current FWER $= \hat{\delta}|\mathcal{H}_{\text{testable}}|$, it is necessary to decrease $\hat{\delta}$ as long as the target FWER is not yet met. Once, the condition $\hat{\delta}|\mathcal{H}_{\text{testable}}| \leq \alpha$ is fulfilled, some patterns that are now untestable under $\hat{\delta}$ must be removed from $\mathcal{H}_{\text{testable}}$.

This necessitates computing the minimal p-value for shapelets at all thresholds. To mitigate this computational burden, we propose a pruning algorithm that allows us to terminate processing a shapelet candidate early. More precisely, the following procedure is based on the insight that we can fill the contingency table of a given candidate/threshold pair step-by-step and abandon future computations once we are certain a p-value lower than or equal to the current $\hat{\delta}$ cannot be reached. Assume we have a partially filled contingency table (i.e. we computed the occurrence of ζ in $u \ll k$ time series). We can bound the p-values of all future table configurations by analysing the two most extreme scenarios:

1. All remaining time series from the positive class have $d(\zeta, T) \leq \theta$ **and** all remaining time series from the negative class have $d(\zeta, T) > \theta$.

2. All remaining time series from the positive class have $d(\zeta, T) > \theta$ **and** all remaining time series from the negative class have $d(\zeta, T) \leq \theta$.

From a partially filled contingency table, we can compute the number of unprocessed time series in the following way: the number of yet to process time series from the positive class is $\Delta_1 = n_1 - a_{\mathcal{S}} - b_{\mathcal{S}}$, and $\Delta_0 = n_0 - c_{\mathcal{S}} - d_{\mathcal{S}}$ for the negative class. For both cases, we have $a_{\mathcal{S}} + b_{\mathcal{S}} + c_{\mathcal{S}} + d_{\mathcal{S}} < n$. If the the minimum p-value of both scenarios exceeds the current significance threshold $\hat{\delta}$, we can stop filling the contingency table and deem the current pattern (subsequence/threshold pair) not significant. We can now express the χ^2 test statistic as a function of $a_{\mathcal{S}}$ and $d_{\mathcal{S}}$ from a partially-filled contingency table:

$$f_{\chi^2}(a_{\mathcal{S}}, d_{\mathcal{S}}) = \frac{n(a_{\mathcal{S}} d_{\mathcal{S}} - (n_1 - a_{\mathcal{S}})(n_0 - d_{\mathcal{S}}))^2}{n_1 n_0 (a_{\mathcal{S}} - d_{\mathcal{S}} + n_0)(d_{\mathcal{S}} - a_{\mathcal{S}} + n_1)}. \tag{2.28}$$

Our contingency table pruning strategy is based on the following theorem:

Theorem 2.1. The maximum of $f_{\chi^2}(a_S, d_S)$ is reached at the boundary of its domain:

$$\max f_{\chi^2}(a_S, d_S) = \max_{a'_S \in [a_S, a_S+\Delta_1], d'_S \in [d_S, d_S+\Delta_0]} f_{\chi^2}(a'_S, d'_S) \tag{2.29}$$

$$= \max\left(f_{\chi^2}(a_S + \Delta_1, d_S + \Delta_0), f_{\chi^2}(a_S, d_S)\right) \tag{2.30}$$

Proof. Both arguments of f_χ^2 are defined on a compact domain, from which follows that under the multivariate generalisation of the extreme value theorem, its extrema lie within the domain or on its boundary. By calculating the partial derivates and setting them to zero, we get solutions of the form $a_S = t$, $d_S = -(tn_0 - n_0n_1)/(n_1)$, for $a_S \in [0, n_1]$. By analysing the determinant of the Hessian, we find that the trivial solutions $a_S = 0$, $d_S = n_0$ and $a_S = n_1$, $d_S = 0$ are (local) minima. Thus, the function's maxima lie on the boundary. From all four boundary cases, it is sufficient to consider the ones that are equivalent to the two scenarios described earlier:

1. $a'_S := a_S + \Delta_1, d'_S = d_S + \Delta_0$
2. $a'_S := a_S, d'_S = d_S$

Both other cases can be neglected, as their test statistic can be increased by decreasing b_S or c_S. ■

From this follows that the minimum p-value that can be obtained from a partially filled contingency table is

$$p'^{\min}_S := 1 - F_{\chi^2}(\max\left(f_{\chi^2}(a_S, d_S)\right)) \tag{2.31}$$

and we can derive the following rule: We stop filling a partially filled contingency table if $p'^{\min}_S > \hat{\delta}$. This allows us to save distance computations and prune a candidate/threshold pair if the condition is fulfilled. Furthermore, if for all thresholds of a given candidate, the respective contingency tables were pruned, we can ignore this candidate altogether, as it will never be testable. Algorithm 1 embeds this procedure into the shapelet mining process and provides a description of the overall `S3M` algorithm. In the next paragraph, we will detail the individual steps of our proposed algorithm.

Once $\hat{\delta}$ and $\hat{\alpha}$ are initialised to 1, a list of all minimum attainable p-values for the data set is calculated. As shown by Llinares-López and Borgwardt [157], $p^{\min}_S(r_S)$ is symmetric around $k/2$ which makes it sufficient to compute all minimum p-values up to $r_S = \lfloor \frac{k}{2} \rfloor + 1$. Then, we iterate over all shapelet candidates and create an empty list of contingency tables for each of them (Line 6). The `UPDATE` routine takes the initial set of contingency tables and the distance between current candidate ζ and time series T as input. For each new distance value (or decision threshold), a new contingency table is added to $\mathcal{C}$, and all other tables are

updated (Line 32). Subsequently, the minimum p-value of the partially filled contingency table $p'^{\min}_{S}$ is computed according to Theorem 2.1. This is done in the `BOUNDARY` routine, which also computes the p-value of both extreme scenarios described earlier. If the minimum of both scenarios exceeds the current significance threshold $\hat{\delta}$, the respective contingency table is rejected. If there are no contingency tables left after processing a given time series, the current candidate will never become significant, no matter how many time series there are left to process, and it can be discarded (break in Line 9). As the number of currently testable patterns has been updated after finishing the loop in Line 7, Tarone's update procedure is called and both the set of testable shapelets $\mathcal{G}$ and significance threshold $\hat{\delta}$ are updated. Once all candidates are processed and the final number of testable patterns is determined, a last mining run is necessary to return the significant shapelets (Line 17). To do so, a single pass over the list of testable patterns is necessary in which shapelets whose *actual* (not minimal) p-value exceeds δ_{Tar} are discarded.

Algorithm 1 S3M (Statistically Significant Shapelet Mining)

Input: Data $\mathcal{D}$, target FWER α

Output: Significant shapelets $\mathcal{G}$, corrected significance threshold δ_{Tar}

procedure S3M($\mathcal{D}, \alpha$)
    Initialize **global** $\hat{\delta} = 1$, **global** $\widehat{\alpha} \leftarrow 1$, and $\mathcal{G}$ to be empty.
    $\mathcal{P} \leftarrow$GENERATE_ALL_MIN_P_VALUES($|\mathcal{D}|$)
    for shapelet candidate $\zeta \in \mathcal{M}$ **do**
        // 1. Contingency table pruning
        $\mathcal{C} = \emptyset$ // Initialise empty list of contingency tables for current candidate
        for Time series $T \in \mathcal{D}$ **do**
            UPDATE($\mathcal{C}, d(\zeta, T), \zeta$)
            break if $\mathcal{C}$ is empty
        end for
        // 2. Tarone's correction procedure [257]
        $\mathcal{G}, \hat{\delta} \leftarrow$ TARONE($\mathcal{P}, \mathcal{G}, \zeta$)
    end for
    // Set final significance threshold
    $\delta_{\text{Tar}} \leftarrow \hat{\delta}$
    // Evaluate *actual* p-value for all patterns in $\mathcal{G}$
    Remove non-significant patterns from $\mathcal{G}$
    return $\mathcal{G}, \delta_{\text{Tar}}$
end procedure

procedure GENERATE_ALL_MIN_P_VALUES(k)
    $\mathcal{P} = \emptyset$
    // All minimum attainable p-values for a data set of size k are symmetric around $k/2$
    for $r_{\mathcal{S}} \in [0, ..., \lfloor \frac{k}{2} \rfloor + 1]$ **do**
        $\mathcal{P} \leftarrow \mathcal{P} \cup p_{\mathcal{S}}^{\min}(r_{\mathcal{S}})$ following Equation 2.24
    end for
    Sort $\mathcal{P}$ in descending order
    return $\mathcal{P}$
end procedure

procedure UPDATE($\mathcal{C}, d, \zeta$)
    Update all contingency tables $\mathcal{C}$ that belong to candidate ζ with d
    for $C \in \mathcal{C}$ **do**
        // Compute minimum attainable p-value for partially-filled C (Equation 2.31)
        $p_{\text{min}} \leftarrow$ BOUNDARY(C)
        // If current candidate/threshold pair is not testable, prune $\mathcal{C}$
        if $p_{\text{min}} > \hat{\delta}$ **then**
            Remove C from $\mathcal{C}$
        end if
    end for
end procedure

Algorithm 2 Tarone's update routine and table pruning

procedure TARONE($\mathcal{P}, \mathcal{G}, \zeta$)
  // Add candidate to list of testable patterns
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{\zeta\}$
  // Update FWER estimate
  $\widehat{\alpha} = \hat{\delta} \cdot |\mathcal{G}|$
  if $\widehat{\alpha} > \alpha$ **then**
    repeat
      // Decrease significance threshold
      $\hat{\delta} \leftarrow$ next value from $\mathcal{P}$
      // After updating $\hat{\delta}$, some patterns in $\mathcal{G}$ are no longer testable
      Remove untestable patterns from $\mathcal{G}$
      // Update FWER estimate
      $\widehat{\alpha} = \hat{\delta} \cdot |\mathcal{G}|$
    until $\widehat{\alpha} \leq \alpha$
  end if
  return $\mathcal{G}, \hat{\delta}$
end procedure

procedure BOUNDARY(C)
  // Create contingency tables for scenarios described in Theorem 2.1
  Fill C_{opt} with remaining $T \in \mathcal{D}$ in a_S and d_S
  Fill $C_{\bar{\text{opt}}}$ with remaining $T \in \mathcal{D}$ in b_S and c_S
  // Compute p-value for both scenarios
  return $\min\{p(C_{\text{opt}}), p(C_{\bar{\text{opt}}})\}$
end procedure

2.4 S3M for Sepsis Detection

Sepsis is a dysregulated host response to an infection that can lead to life-threatening organ dysfunction [237]. While being one of the most common causes of in-hospital death, the early recognition of sepsis and timely treatment interventions remain unsolved challenges in the biomedical domain. As pointed out in a recent review by Moor et al. [178], data-driven biomarker discovery remains an open challenge, which may improve our understanding of sepsis. In this section we will use S3M to extract and analyse temporal biomarkers that are

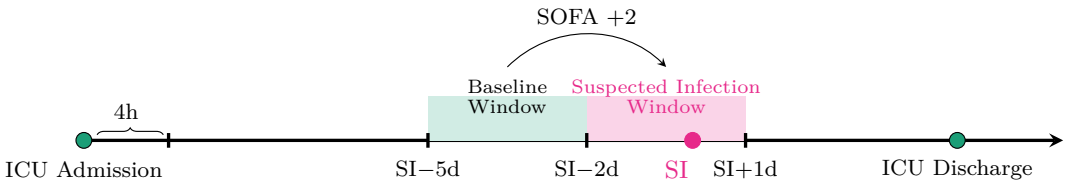


Figure 2.2: Adapted Sepsis-3 definition [237] as used in our study. A sepsis case shows a Sequential Organ Failure Assessment (SOFA) score [274] increase of at least two when comparing a baseline window to a window around a suspected infection (SI).

statistically associated with sepsis from the MIMIC-III database [124]. In the following sections, we will use the term "significant" to express *statistical* significance.

2.4.1 Data Set & Preprocessing

The MIMIC-III data set [124] contains data from over 50 000 intensive care unit (ICU) stays arising from the continuous monitoring of over 45 000 critically ill patients. The database queries described in the following paragraphs are based on the MIMIC Code Repository [123].

Cases To extract septic patients from MIMIC-III, we apply an adapted form of the Sepsis-3 definition by Singer et al. [237] as visualised in Figure 2.2: if both of the following two criteria are met, we identify a patient as a case:

1. An antibiotic was administered and a sample of body fluid cultures was taken. These actions by a doctor indicate a suspected infection (SI).

2. The Sequential Organ Failure Assessment (SOFA) score increases by two points when comparing two maxima: one from a window five to two days before SI; one from a window between two day before to one day after the suspected infection.

Additionally, we require that the SI occurred at least four hours into the ICU stay.

Controls Any patient without a septic period is part of the control cohort. This definition allows that *one* of the aforementioned conditions may be met and the patient would still be considered a control. Furthermore, even patients with a septic period before or immediately after the ICU stay may be included in the control group. This more inclusive control definition was a deliberate choice to find biomarkers that distinguish sepsis cases not only from comparably "healthy" individuals but from a heterogeneous control cohort.

We excluded patients whose vitals were logged with the CareVue system as it was shown [70] that it under-reports crucial negative microbiology lab values. Furthermore, we excluded young patients who are below the age of 15 as physiological responses in paediatric sepsis are vastly different from adult sepsis cases [170].

With these criteria, we identified 355 cases (0.58 %) and 21 079 controls. To receive a balanced data set, the control cohort was downsampled to be of the same size as the case cohort. We extracted three vital signs that are routinely monitored in ICU patients and are important indicators of patient stability: Heart rate, respiratory rate, and systolic blood pressure. Aperiodically measured parameters were forward and backward filled to achieve an harmonised sampling rate of 30 minutes. Lastly, we used only the first 75 hours of a patient stay for time series extraction to generate shapelet candidates from a diverse set of patients (instead of extracting the majority of candidates from few very long stays).

2.4.2 Experimental Setup

Each set of vitals is divided into 66 % training, and 33 % test set. `S3M` is used to extract shapelets on the training set, on which we also extract their p-values. In all analyses pertaining to classification performance, assessments are made on the test set. Before running our method, we remove duplicate candidates (i.e. identical subsequences) from $\mathcal{M}$ as we do not want to test the same hypothesis multiple times.

Parameters We mined statistically significant subsequences of lengths $w_0 = 4, w_1 = 5, w_2 = 6$ which results in sequences of 2, 2.5, and 3 hours of length, respectively. While longer periods are possible, domain experts deemed these shorter lengths more useful in a diagnostic setting. The target FWER was set to 0.01, a significance threshold commonly used in the literature. We also reduced the number of cases and controls for the systolic blood pressure data to 75 each. This was due to the observation that many blood pressure time series are not sufficiently different from each other leading to many virtually interchangeable shapelet candidates.

Comparison Partner `S3M` is virtually the first shapelet-based approach which makes use of a statistical selection criterion. This allows us to first mine shapelets that are statistically associated with sepsis (i.e. subsequences of high descriptive power w.r.t. the phenotype), and second, assess their classification performance using the threshold rule described at the end of Section 2.1.1. We therefore select a shapelet-based baseline method that has been particularly successful in the classification setting and compare p-values and predictive performance of both methods. Karlsson et al. [127] introduced Generalized Random Shapelet

Forests (gRSF), an efficient random forest based algorithm for time series classification. It constructs decision trees based on randomly selected shapelets extracted from a randomly selected subset of training time series. Trained in the same way as a classical random forest [38], it yields a set of selected shapelets that maximise predictive performance. This allows us to compute their p-values and determine whether the algorithm selects any statistically significantly associated shapelets.

2.4.3 Results

In the following sections, we report two types of results. First, we perform a statistical analysis contrasting statistical association of the shapelets that our method identified with the ones form a competing method. We also illustrate, that `S3M` can be used to find subsequences that provide competitive classification performance. Second, we provide a medical analysis and interpretation of detected shapelets.

2.4.3.1 Statistical Analysis

In a first analysis, we take a look at the number of significant shapelets detected by `S3M` and its significance threshold on the training data set. We contrast our counts with the shapelets used by gRSF for classification under the respective threshold δ_{gRSF}, and show the naïve Bonferroni-corrected threshold δ_{Bon}. The latter differs between identically-sized data sets (heart rate and respiratory rate) because we employ a duplicate-removal-step prior to running `S3M`. This decreases the number of tested hypotheses (see Equation 2.19 on page 28) and leads to a data-set-dependent correction factor. Table 2.2 shows that compared to gRSF, using Tarone's adjustment procedure [257] leads to a much higher (less conservative) significance threshold (δ_{Tar}) and a high number of significant shapelets. This is due to the fact that during the creation of each decision tree in the random forest, a random shapelet is selected for each node in the tree leading to many implicitly tested hypotheses for which a correction is required. This search-space inflation results in the observation that no shapelet used by gRSF is significant, emphasising that high predictive performance does not imply statistical association. When comparing to the naïve Bonferroni correction δ_{Bon}, we see that Tarone's procedure results in a slightly less conservative significance threshold. More precisely, for the vital signs heart rate, respiratory rate, and systolic blood pressure, our method determines that 34.69 %, 72.03 %, and 39.77 %, of all hypothesis are untestable. In Table 2.3, we empirically observe that utilising significant shapelets for classification leads to competitive prediction performance. However, more generally, it is neither guaranteed that statistical association implies high predictive performance nor that high accuracy implies a statistical

Table 2.2: Number of significant shapelets detected by S3M and gRSF as well as significance thresholds. The significance threshold reached by our method is denoted as δ_{Tar}, the Bonferroni correction factor by δ_{Bon}. Note that despite identical data set sizes (heart rate and respiratory rate), δ_{Bon} differs due to the removal of duplicate shapelet candidates before the mining process. Note also that for the computation of δ_{gRSF}, all possible candidates must be considered, even if some have been pruned, as they are still tested implicitly.

Vital Sign	S3M	gRSF	δ_{Tar}	δ_{gRSF}	δ_{Bon}
Heart Rate	200	0	2.51×10^{-10}	1.28×10^{-15}	1.87×10^{-10}
Respiratory Rate	514	0	4.47×10^{-10}	1.33×10^{-15}	2.10×10^{-10}
Systolic Blood Pressure	58	0	2.55×10^{-9}	4.35×10^{-14}	1.97×10^{-9}

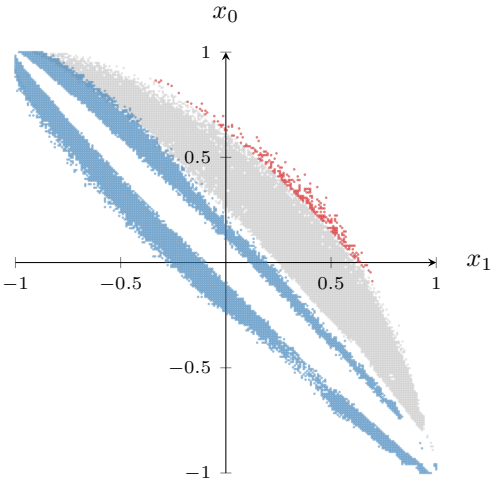
association. In this context it is important to note that association mapping and classification are related supervised tasks, which differ, however, in their objectives. The aim of finding shapelets that maximise predictive performance is to build a reliable predictor for unseen data, whereas association mapping is a hypothesis generation tool. A statistically associated shapelet may be of limited predictive but high scientific value as it facilitates a better understanding of the mechanisms behind an outcome. Here, we evaluate significant shapelets on the test set and report the highest accuracy score to show the potential of a statistics-driven shapelet selection. For gRSF, we average its results from ten runs and observe that a *single* shapelet from S3M may lead to a classification performance on a level that is comparable to the combination of over 3000 in a random forest.

An in-depth analysis of all contingency tables of the heart rate and blood pressure data sets is depicted in Figure 2.3. Each axis represents the degree to which a shapelet is present in the cases (x_1) and absent in the controls (x_0). More precisely, expressed in terms of the contingency table counts (see Table 2.1), we have

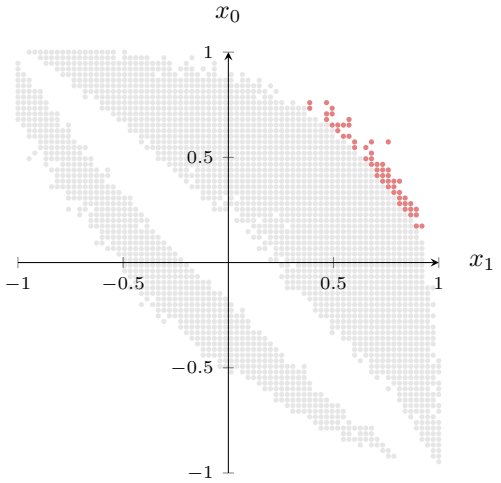
$$x_1 = \frac{a_S - b_S}{a_S + b_S} \quad \text{and} \quad x_0 = \frac{d_S - c_S}{d_S + c_S}.$$

Table 2.3: Classification accuracy of S3M versus gRSF on the test set. Reproduced from [26] with permission from Oxford University Press.

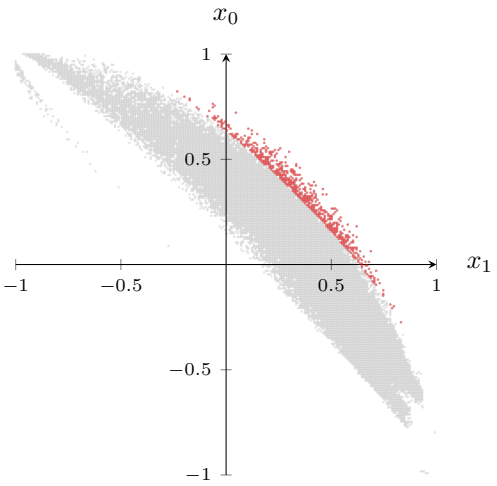
Vital Sign	S3M	# shapelets	gRSF	# shapelets
Heart Rate	0.70	1	**0.74**	3030
Respiratory Rate	0.71	1	**0.76**	3406
Systolic Blood Pressure	**0.75**	1	0.74	971



(a) Contingency table visualisation of shapelets from the **heart rate** data set. Blue dots are only visualised to show a "null distribution" of contingency tables with randomly permuted class labels.



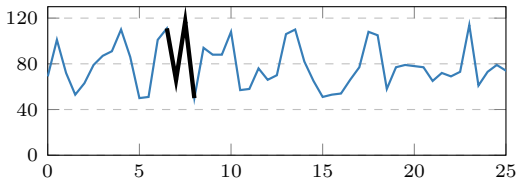
(b) Contingency table visualisation of shapelets from the **blood pressure** data set.



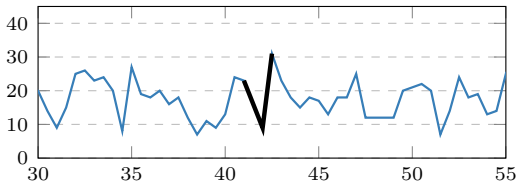
(c) Contingency table visualisation of shapelets from the **respiratory rate** data set.

Figure 2.3: Contingency table plots summarising the distribution of shapelets in cases and controls. Red dots show *statistically significant* shapelets. Non-significant shapelets are marked in grey. The distribution of candidate shapelets under label permutation are shown in blue. Reproduced from [26] with permission from Oxford University Press.

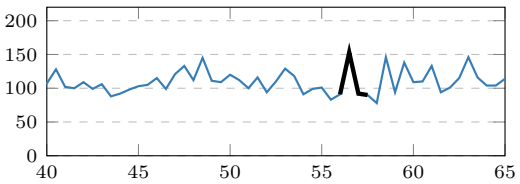
Intuitively, a shapelet that perfectly distinguishes both classes (e.g. it occurs in all cases and never in any control) will have coordinate $(x_1 = 1, x_0 = 1)$. As each dot represents a contingency table configuration, we mark significant shapelets whose p-value falls under the significance threshold δ_{Tar} in red and all others in grey. As a "visual null distribution", we also plot contingency tables from the heart rate data set when randomly permuting the class labels (only shown in Figure 2.3a). In all data sets significant shapelets are distinctly scattered in the first quadrant. Showing that statistically significant shapelets are predominantly present in *cases* which implies that we found subsequence that may help identify a septic patient (rather than a signal that identifies a healthy one).



(a) Heart Rate (beats per minute)



(b) Respiratory Rate (breaths per minute)



(c) Systolic Blood Pressure (mmHg)

Figure 2.4: Most significant shapelets (on the training set) in their time series of origin. Reproduced from [26] with permission from Oxford University Press.

2.4.3.2 Medical Interpretation

While the following observations highlight overlaps between shapelets and established and well known manifestation of sepsis, their interpretations need to be substantiated by further research. Our method should be seen as a hypothesis generation or confirmation tool that can support researchers with the analysis of biomedical time series data sets.

Shapelet-based methods are by construction interpretable, a property that makes them particularly interesting for hypothesis generation. Paired with a statistical interpretation, S3M can support the generation of deeper domain-specific insights. Figure 2.1 visualises the most significant shapelets detected on the training set by S3M and the respective time series of origin.

In the upper plot, we detected a shapelet showing a transient instability of the heart rate. While this seems to contradict the observation that low heart rate variability (HRV) correlates with higher sepsis severity [4, 49], it is important to point out the following details: In the context of biomedical research, HRV refers to the variance of the temporal distance between *individual* heart beats. This definition provides a very high sampling resolution and is not comparable to our setting in which we consider the number of heart beats per minute on a grid of 30 minute intervals. The signal captured in Figure 2.4a is therefore different from the common notion of HRV and highlights the possible importance of low-frequency HRV for sepsis (e.g. due to haemodynamic instability).

A well established link between the pulmonary system and sepsis manifests itself in an increased respiratory rate (RR) [115]. This may be due to pulmonary oedema, lactate acidosis, or other pathomechanisms. More precisely, an RR of 20 breaths per minute or more constitutes one of four factors of the systemic inflammatory response syndrome (SIRS), a list of criteria relevant for an earlier definition of sepsis [30]. It is therefore interesting to observe that the shapelet in Figure 2.4b contains two abnormal measurements (at both ends) drawing attention to a particularly high rate of 31 breaths per minute. Lastly, the shapelet detected in systolic blood pressure is visualised in Figure 2.4c. It contains a distinct spike into an abnormally high regime indicating an overall poor state of health.

2.5 Conclusion

In this section, we addressed the problem of identifying statistically significant time series patterns in a scalable manner. Our method utilises short subsequences (shapelets) and assesses their statistical significance by association testing. To mitigate the multiple hypothesis testing problem, Tarone's method [257] is employed to improve run-time by pruning untestable shapelets. Moreover, we exemplified the merit of our method on a data set containing vital signs of patients that suffer from sepsis. Detected shapelets are biomedically relevant as they can serve as data-driven medical hypotheses whose importance can be further investigated by clinical researchers. In addition, we demonstrated that statistically associated shapelets may also be used in a classification setting with promising results on predictive performance. This direction was explored in a follow-up work that uses `S3M` in an early-detection system of circulatory failure in the ICU [117]. In combination with additional time series features, we were able to predict 90 % of circulatory-failure events identifying 82 % more than 2 h in advance.

A challenge that arises from our exhaustive search is twofold. First, redundant and overlapping shapelets may be discovered. Second, enumerating all candidates leads to high mem-

ory consumption. The first issue can lead to an unbounded number of structurally meaningless shapelets which impacts the practically of S3M. In a second follow-up work [97], we introduced Statistically Significant Submodular Subset Shapelet Mining (S5M), a method that maintains structural diversity of shapelets during the mining process using submodular optimisation. This way, detected shapelets are 1. representative of the data set 2. minimally redundant, and 3. more manageable in cardinality, increasing S5M's practicality and statistical power. Regarding the problem of space complexity, symbolic encodings such as the Symbolic Aggregate approXimation (SAX) [153] of shapelet candidates could dramatically reduce the size of the search space. Simultaneously, SAX would prevent the extraction of candidates that are morphologically similar as their approximations would be identical. Lastly, learning shapelets without resorting to enumeration (e.g. through dictionary [295] or gradient-based [95] learning) may be used to mine shapelets in a generative manner to further reduce the computational burden.

3 Time Series Classification

> In which we develop a novel classification algorithm using shapelets, followed by the development of a deep learning system to predict myocardial ischaemia.

In Chapter 2, we used shapelets to find subsequences that are statistically associated with a class or phenotype of interest. We also highlighted the differences and commonalities of significantly associated subsequences and subsequences that maximise predictive performance. In this chapter, we consider the problem of time series classification (TSC), a task that is complementary to association mapping. This chapter is organised as follows: First, we will use the same subsequence extraction method as in Chapter 2 to develop a new general-purpose TSC algorithm. We will do so by using concepts from optimal transport theory to define a similarity measure based on the subsequence representation of a time series. For this, we will first provide a brief introduction to kernel methods and optimal transport, followed by the description of our proposed method. Section 3.2 is based on the following publication:

- **C. Bock**[†], M. Togninalli[†], E. Ghisu, T. Gumbsch, B. Rieck, and K. Borgwardt. "A Wasserstein Subsequence Kernel for Time Series". In: *2019 IEEE International Conference on Data Mining (ICDM)*. 2019, pp. 964–969. DOI: 10.1109/ICDM.2019.00108

From the general-purpose approach, we then move to a more specific classification problem in healthcare. Section 3.3 is based on work in progress in which a deep learning based system for the identification of exercise-induced myocardial ischaemia is developed. It first introduces the data type (electrocardiogram) and problem we are concerned with before giving a brief overview of recent advances in deep learning for cardiology. Subsequently, we detail the data set, experimental setup, and the employed neural network architecture. In the result section, we report predictive performance on an internal held-out test set and its clinical relevance as well as aspects of trust and interpretability.

3.1 Introduction

Time series classification is the task of predicting the class membership of a given time series. It is an active research topic applicable to various fields such as clinical event prediction [178, 199, 248] and human activity [185] or gesture recognition [172]. Over the last years, hundreds of general-purpose classification algorithms have been introduced and were evaluated on the central repository for time series classification benchmark data: The "UCR Time Series Archive" [66]. These methods were systematised by Bagnall et al. [12, 216] who defined the following taxonomy to differentiate between six types of TSC approaches:

1. *Whole series* approaches use a measure of similarity over the complete time series (e.g. Dynamic Time Warping (DTW) [20]) that can subsequently be used in a distance-based classification algorithm.

2. *Interval-based* methods select one or more phase dependent intervals for comparison.

3. *Shapelet-based* algorithms aim to find phase-*independent* patterns whose occurrence in a time series is indicative of a specific class label.

4. *Dictionary-based* approaches use frequency counts of recurring patterns in forms of histograms as feature representation.

5. *Combinations* of two or more procedures from above.

6. *Model-based* algorithms use the similarity of generative models fit to each time series individually as a proxy of time series similarity.

Furthermore, deep learning methods have been increasingly popular in the TSC community [80] due to their revolutionising impact in fields such as computer vision and natural language processing. Successful deep learning approaches for time series classification make use of convolutional neural networks (CNN) [145], residual networks [105], long short-term memory (LSTM) [108] recurrent neural networks (RNN) [217], and the more recently developed attention mechanism [270].

In the next sections, we will first introduce a novel general-purpose TSC algorithm that belongs to the *shapelet-based* class of algorithms. Subsequently, we will conclude the first part of this thesis by developing a classification system based on deep learning for the identification of myocardial ischaemia.

3.2 Subsequence Kernels for Time Series Classification

In Section 2.1, we highlighted the importance of time series subsequences as general feature descriptors for time series classification. Subsequences that maximise predictive performance are referred to as "shapelets". In the sections that followed, we repurposed shapelets and developed a method that allows to efficiently assign p-values to subsequences enabling the discovery of interpretable and descriptive temporal biomarkers. In what follows, we will utilise subsequences and their distributions to develop a novel method for time series classification. The method's core is a kernel function defined over the Wasserstein metric, a distance measure rooted in optimal transport (OT) theory.

3.2.1 Kernel Methods

The most prominent representative of kernel methods is the Support Vector Machine (SVM) [34]. One of the powerful properties of SVMs is the fact that any binary classification task can be expressed as a *constrained convex minimisation problem* ([227, Corollary 6.6]). In this context, either all solutions to the minimisation problem are equally good, or there is *one unique* solution to the problem [227, Chapter 6.1]. An integral part of the SVM and kernel methods as a whole is that they require an inner product that measures the similarity of two objects. This inner product arises in the shape of a symmetric positive semi-definite (PSD) kernel function k defined over the input space: $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For every $c_i \in \mathbb{R}$ and $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, we have

$$\sum_{i,j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \tag{3.1}$$

where n is the number of samples of the input space. Note that Schölkopf and Smola [227, Remark 2.6] point out that k as presented here may also be referred to as positive definite (PD), although in matrix theory, the term *definite* is reserved for cases in which Equation 3.1 holds only if $c_1 = \cdots = c_n = 0$.

Given the non-empty set $\mathcal{X}$, positive semi-definite k, and a function $f\colon \mathcal{X} \to \mathbb{R}$, we can show that there is a space $\mathcal{H}$ in which the output of k can be expressed as an inner product. We say that $\mathcal{H}$ is a Hilbert space on $\mathcal{X}$ if the following two properties are fulfilled.

1. $k(\cdot, \mathbf{x}) \in \mathcal{H}$ for all $\mathbf{x} \in \mathcal{X}$, and mutatis mutandis $k(\mathbf{x}, \cdot) \in \mathcal{H}$ for all $\mathbf{x} \in \mathcal{X}$

2. $f(\mathbf{x}) = \langle f, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$ and $\mathbf{x} \in \mathcal{X}$

If this is the case, k is the *unique* reproducing kernel of $\mathcal{H}$, and we can write

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle k(\cdot, \mathbf{x}_i), k(\mathbf{x}_j, \cdot) \rangle_{\mathcal{H}}. \tag{3.2}$$

Moreover, the Moore–Aronszajn theorem [10] states that every PSD function is the reproducing kernel (according to Equation 3.2) of some *unique* Hilbert space $\mathcal{H}$. In other words, for all PSD functions k, there exists a Hilbert space $\mathcal{H}$ such that Equation 3.2 holds and k is *reproduced* in that space. Due to this property we refer to $\mathcal{H}$ as reproducing kernel Hilbert space (RKHS). As another consequence of the theorem from above, we may consider the following concepts to be equivalent:

- Kernel functions
- Reproducing kernels
- Positive semi-definite functions

$\mathcal{R}$-convolution Kernels To extend the construction of kernels to discrete structures such as strings or graphs, Haussler [104] introduced the $\mathcal{R}$-convolution kernel framework. Given an object $x \in \mathcal{X}$, we assume it can be decomposed into a set of D structures $\{x_1, \ldots, x_D\}$ each of which has a "is part of" relationship to x. We denote this relationship as the relation $R(x_1, \ldots, x_D, x)$ that is true iff each x_d is a part of x. Conversely, $R^{-1}(x)$ is the set of all elements that are "part of" x: $R^{-1}(x) = \{x_d | R(x_d, x)\}$. An intuitive example of viewing an object as a composite structure is the decomposition of a time series into its subsequences as introduced in Section 2.1.1.

More generally, we assume x_d to be in the set $\mathcal{X}_d$ for each $1 \leq d \leq D$, where $\mathcal{X}_1, \ldots, \mathcal{X}_D$ are a non-empty, separable metric spaces. Let us now collect all decompositions of $x, y \in \mathcal{X}$ in $\vec{x} = x_1, \ldots, x_D$ and $\vec{y} = y_1, \ldots, y_D$. Furthermore, let $k_d(x, y)$ be a kernel on $\mathcal{X}_d$ which measures the similarity of part x_d and y_d and define the generalised convolution as

$$k(x, y) = \sum_{\vec{x} \in R^{-1}(x)} \sum_{\vec{y} \in R^{-1}(y)} \prod_{d=1}^{D} k_d(x_d, y_d). \tag{3.3}$$

The $\mathcal{R}$-convolution of $k_1, \ldots, k_D$ is defined to be the zero extension of k to $\mathcal{X} \times \mathcal{X}$, and denoted as $k_1 \star \cdots \star k_D(x, y)$. Haussler proved in [104] that if $k_1, \ldots, k_D$ are kernels on $\mathcal{X}_1 \times \mathcal{X}_1, \ldots, \mathcal{X}_D \times \mathcal{X}_D$, and if R is a finite relation on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_D \times \mathcal{X}$, then $k_1 \star \cdots \star k_D(x, y)$ is a kernel on $\mathcal{X} \times \mathcal{X}$.

In its most simple instantiation, the $\mathcal{R}$-convolution kernel can be expressed as the sum of a base kernel evaluated for all combinations of all parts of two objects x and y, i.e.

$$k(x, y) = \sum_{a \in R^{-1}(x)} \sum_{b \in R^{-1}(y)} k_{\text{base}}(a, b), \tag{3.4}$$

where the base kernel k_{base} is any valid kernel function defined over the parts of the objects at hand. In Section 3.2.3, we will show that for a subsequence decomposition of time series and a linear base kernel this framework is downright meaningless and develop a new kernel based on OT.

Kernel Methods for Time Series Classification One of the first kernel methods for time series classification does not make use of the $\mathcal{R}$-convolutional framework. Instead, Rüping [218] uses "standard" SVM kernels (such as linear and radial basis function (RBF) kernels) as a way to compare *whole* time series. The classification of time series that express periodic patterns was investigated using several cross-correlation kernels by Wachman et al. [275]. In addition, Lorincz et al. [161] introduced a set of methods that use DTW, an alignment-based similarity measure for time series to classify emotional expressions from facial landmark positions. However, in general, DTW-based kernels do not fulfil the condition of Equation 3.1 which started an investigation into the impact of such "indefinite" kernels. This culminated in the recursive edit distance kernel presented by Marteau and Gibet [169]. Furthermore, Cuturi at al. [58, 63] showed that an alignment-based PSD kernel can be derived by taking the softmax over all possible alignments. For variable-length multivariate time series, the same author [62] followed the idea of vector autoregressive (VAR) models to define autoregressive kernels as an instance of covariance kernels.

Closest to the approach we will introduce in the following sections is the kernel earth mover's distance (`KEMD`) [65]. While the authors also build on optimal transport theory (the earth mover's distance [214] is equivalent to a certain instance of the Wasserstein metric), the ground distance matrix is constructed using histograms of time series observations. This histogram intersection kernel [186] treats each time series as a one-dimensional distribution of scalars, whereas our approach uses the distance between high-dimensional distributions of *subsequences* to construct the ground distance matrix. We empirically show that by using subsequences, our approach captures long-distance similarities in time series better.

3.2.2 Optimal Transport (OT)

Optimal transport theory is a field of mathematics which investigates problems in resource allocation and transportation as illustrated in the the following example (inspired by the

famous Hitchcock problem [107]): Suppose there are n cheeseries (sources) distributed all over the Swiss alps, each of which with a different production capacity. Furthermore, let there be m huts (sinks) that have a certain demand of cheese that needs to be satisfied. Lastly, we define a cost function describing the cost of transportation of a wheel of cheese from a cheesery to a hut (e.g. the spatial distance between cheesery and hut). Informally, the aim of OT is to find a transportation plan that minimises the overall transportation cost while fulfilling the hut's demands and considering the production capacity of all cheeseries. In other words, we would like to know how many wheels a cheesery should delivery to any given hut such that the travelled distance of *all* wheels is minimal.

In this scenario, we dealt with discrete quantities but in order to formalise the problem in a more general manner, we will make use of probability distributions as source and sink quantities, before coming back to the discrete case. A concept intrinsically linked to the search for the optimal transport plan is the so-called p-Wasserstein distance. Given two probability distributions σ and μ defined on a metric space $\mathcal{M}$ and a ground distance function $\text{dist}(\cdot, \cdot)$, we have:

Definition 3.1. p-Wasserstein distance. For some $p \geq 1$ and the set of all possible transportation plans $\Gamma(\sigma, \mu)$ over $\mathcal{M} \times \mathcal{M}$ that have marginals σ and μ, the p-Wasserstein distance is

$$W_p(\sigma, \mu) = \left(\inf_{\gamma \in \Gamma(\sigma,\mu)} \int_{\mathcal{M}\times\mathcal{M}} \text{dist}(x, y)^p \mathrm{d}\gamma(x, y) \right)^{\frac{1}{p}}. \tag{3.5}$$

In essence, the Wasserstein distance defined as above measures the minimum cost it takes to transform or morph one probability distribution into another.

In its discrete formulation, and to circle back to the cheese and hut example, we can express the 1-Wasserstein distance also as a minimisation problem over the Frobenius inner product of two matrices. Given two sets of k-dimensional features $X \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{m\times k}$, let D be the $n \times m$ matrix of pairwise distances such that $D_{i,j} = \text{dist}(\mathbf{x}_i, \mathbf{y}_j)$, where $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. Then, we have

$$W_1(X, Y) = \min_{P\in\Gamma(X,Y)} \langle D, P \rangle_F, \tag{3.6}$$

where the Frobenius product is defined as $\langle D, P \rangle_F = \sum_{i,j} D_{i,j} P_{i,j}$. The argument of the solution (the best transportation plan $P \in \mathbb{R}^{n\times m}$) consists of fractions that indicate how to transport values from X to Y. We assume that the total transported mass is equal to 1 and that across all elements of X and Y this mass is evenly distributed. This constrains the rows of P to sum up to $1/n$ and its columns to $1/m$.

3.2.3 Time Series Kernels and Optimal Transport

In this section, we will develop a new method for time series classification combining $\mathcal{R}$-convolution framework with the Wasserstein distance and the shapelet representation from Section 2.1. Before describing and evaluating the method we refer to as Wasserstein Subsequence Kernel (`WTK`), we show that in its most simple application to time series subsequences, the $\mathcal{R}$-convolution framework is de facto meaningless and more expressive representations are needed.

3.2.3.1 Motivation

Consider two time series T and T', and their length-w subsequence sets $\mathcal{M}$ and $\mathcal{M}'$, respectively. Extending slightly on Equation 3.4, we define the following kernel function:

$$k(T, T') = \frac{1}{|T| \cdot |T'|} \sum_{\mathbf{S} \in \mathcal{M}} \sum_{\mathbf{S}' \in \mathcal{M}'} k_{\text{base}}(\mathbf{S}, \mathbf{S}'), \tag{3.7}$$

where the bold notation of a subsequence indicates its representation as a vector. When choosing k_{base} as a linear kernel (i.e. $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$), we have

$$\begin{aligned} k(T, T') &= \frac{1}{|T| \cdot |T'|} \sum_{\mathbf{S} \in \mathcal{M}} \sum_{\mathbf{S}' \in \mathcal{M}'} \mathbf{S}^\top \mathbf{S}' && (3.8) \\ &\approx \frac{1}{|T| \cdot |T'|} \left(\sum_{\mathbf{S} \in \mathcal{M}} \mathbf{S}^\top \right) \left(\sum_{\mathbf{S}' \in \mathcal{M}} \mathbf{S}' \right) && (3.9) \\ &\approx \overline{T}^\top \overline{T'}, && (3.10) \end{aligned}$$

where the bar notation refers to the time series mean. The last approximation follows from observation that in the sums over subsequences, all length-w observations (except the leading/trailing $w - 1$ observations) appear at all dimensions in the sum. From this follows that for many choices of w, the $\mathcal{R}$-convolution kernel with linear base kernel degenerates to the comparison of time series means. In particular for z-normalised data (zero mean and unit variance), a recommended preprocessing step [66, 201], this observation renders the straightforward application of the $\mathcal{R}$-convolution framework for time series classification practically *meaningless*. Figure 3.1 depicts this observation for 4 data sets from the "UCR Time Series Archive" [66] containing only z-normalised data. We show the mean of the kernel matrix on the y-axis and the subsequence length w on the x-axis. As expected, for short subsequences, the mean has a tendency of staying close to zero. However, in Section 3.2.3.3, we demon-

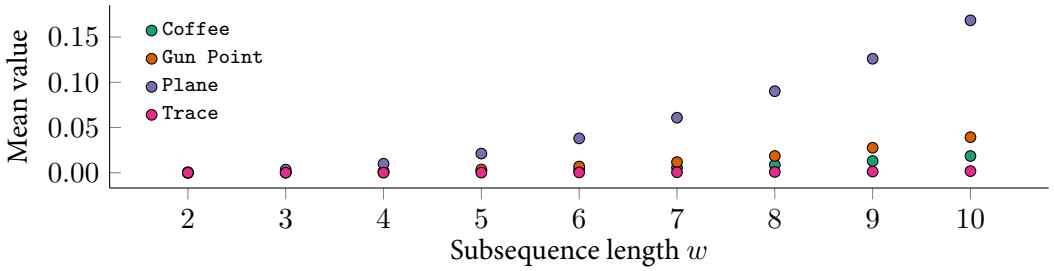


Figure 3.1: The mean value of the kernel matrices of 4 time series classification data sets and varying subsequence lengths w. The kernel is constructed as described in Equation 3.8.

strate that the naïve application of $\mathcal{R}$-convolution framework does *not* lead to competitive accuracies even for longer subsequences.

3.2.3.2 A Wasserstein Subsequence Kernel (WTK)

Following the notation in Section 2.1.1, we are given a time series data set $\mathcal{T}$ with k samples for each of which we extract a set length-w subsequences $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$. Our novel Wasserstein distance based kernel is defined as follows:

Definition 3.2. Wasserstein time series kernel. Given two time series T_i and T_j, let $\mathcal{M}_i = \{\mathbf{S}_{i,1}, \ldots \mathbf{S}_{i,U}\}$ and $\mathcal{M}_j = \{\mathbf{S}_{j,1}, \ldots \mathbf{S}_{j,U}\}$ be the the set of their respective subsequences. Furthermore, let $D \in \mathbb{R}^{U \times V}$ be the distance matrix containing the pairwise Euclidean distances of all subsequences. According to Equation 3.6, we obtain the 1-Wasserstein distance between T_i and T_j by transforming one time series into the other using their subsequence representations by solving the following optimisation problem:

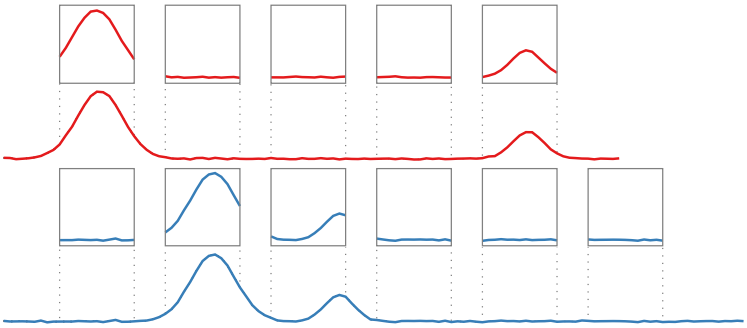
$$W_1(T_i, T_j) = \min_{P \in \Gamma(T_i, T_j)} \langle D, P \rangle_F. \tag{3.11}$$

Given the constant factor $\lambda \in \mathbb{R}_{>0}$, our *Wasserstein subsequence kernel* is defined as

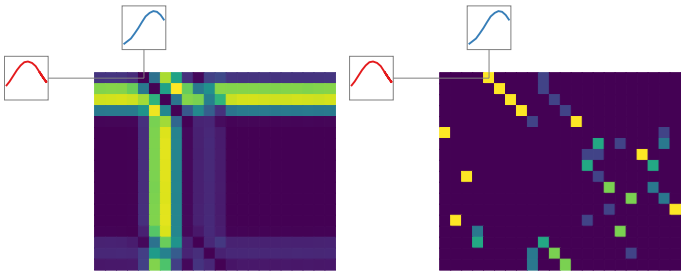
$$\text{WTK}(T_i, T_j) = \exp\left(-\lambda W_1(T_i, T_j)\right). \tag{3.12}$$

For notational simplicity, we will write $\text{WTK}(\mathcal{M}_i, \mathcal{M}_j) := \text{WTK}(T_i, T_j)$ and $W_1(\mathcal{M}_i, \mathcal{M}_j) := W_1(T_i, T_j)$. Lastly, Villani [273] showed that $W_1(T_i, T_j)$ is a metric and could therefore be used in the k-nearest neighbour (k-NN) algorithm.

Intuition WTK belongs to the family of shapelet-based methods that use subsequences as the representation of a time series. Subsequence features are intrinsically interpretable, and



(a) Extraction of subsequences



(b) Subsequence distance matrix (c) Optimal transport plan

Figure 3.2: Given two time series, we compute their subsequence-based Wasserstein distance in multiple steps. (a) illustrates the first step in which subsequences are extracted using a sliding window. Note that we do not show all subsequences because of overlapping windows. We then, use the Euclidean distance to compute distance matrix D (b). Blue indicates small distances, while yellow indicates large distances. Lastly, we compute the optimal transport plan (c). Colours indicate the amount of mass being transported. High (yellow) values can be interpreted as subsequences that are "matched". I.e. the highlighted subsequences are assigned to each other. The transport plan contains fractional matchings as both time series are of different lengths. This enables us to make use of subtle differences in subsequence distributions when computing our distance. ©2019 IEEE

we can therefore visualise each step of our distance computation as depicted in Figures 3.2 and 3.3. This includes the extraction of length-w subsequences (Figure 3.2a), followed by the computation of pairwise distances (Figure 3.2b) and calculations of the final optimal transport plan (Figure 3.2c). From solving the optimisation problem in Equation 3.11, we obtain the transport plan P. As shown in Figure 3.2c, this transport plan assigns each subsequence from time series T_i to *one or more* subsequences from time series T_j. Figure 3.3 depicts the transport plan mapping of subsequences in more detail. The way the optimisation problem is formulated (i.e. the fact that it is not necessary that D in Equation 3.11 is a square matrix) accounts for subsequence sets of different cardinalities. Time series of varying lengths can therefore be processed by our method without any modifications. If both time series are of the same length (as is the case for all data sets of the "UCR Time Series Archive" [66]),

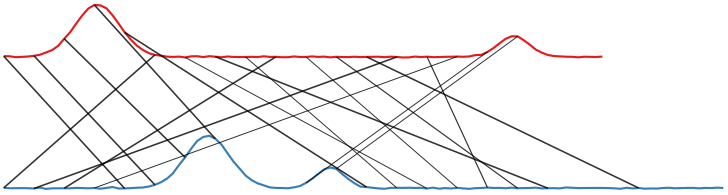


Figure 3.3: An illustration of the optimal transport plan from Figure 3.2c. Each line connects the beginning of two matched subsequences. The transported mass is encoded in the line's thickness. We only show large transport values to avoid cluttering. © 2019 IEEE

the mapping a one-to-one correspondence. To obtain the Wasserstein distance, a sum over the values obtained by element-wise multiplication of both matrices shown in Figures 3.2b and 3.2c is computed. The distance value captures the difference between the time series in terms of their subsequence distributions and is more expressive than merely summing the values of the transport plan alone. In our example, we observe that the optimisation procedure "selects" the lowest subsequence distances and correctly aligns the peaks of both time series.

To summarise, there are three components that define our similarity measure: 1. the way we extract subsequences, 2. the way we define similarity between them, and 3. computing an alignment based on these similarities. By extracting only subsequences of one length and choosing the Euclidean distance to compute the ground distance matrix (as proposed), we assume that no time warping occurs in our data set. We could relax this assumption by using DTW as subsequence distance function and set up the extraction process in a way that sequences of multiple lengths are considered. Our choice of subsequence distance also implies that we consider the order in which patterns occur as not relevant. While in problems such as arrhythmia detection, this is a reasonable assumption, we could take the position of a pattern in the time series of origin into account when setting up the ground distance matrix. Lastly, computing a distance based on the optimal transport plan (i.e. the 1-Wasserstein distance) implies that our application is such that actual pattern matches are essential. While both other components allow us to determine the nature of a pattern, this last component is immutable and the core component of `WTK`.

Theoretical Properties To use our similarity measure from Equation 3.12, we must show that it yields a proper kernel. More specifically, if our kernel should belong to an RKHS, it is necessary to show that it is PSD, i.e. it satisfies Equation 3.1 for all $c_i \in \mathbb{R}$. Feragen et al. [82, Theorem 5] show that the above condition is fulfilled if one can show (conditionally) negative definiteness of the symmetric distance matrix $D_{ij} = W_1(T_i, T_j)$ for any given data

set. This means that D must not have more than *one* positive eigenvalue [14, Lemma 4.1.4, p. 163]. If this was the case, a metric space is induced by Equation 3.11 for which an isometric embedding into a Hilbert space exists.

During the evaluation of our empirical results, some configurations yielded at least two positive eigenvalues in D. From this follows that the kernel matrix $\mathcal{K}_{ij} = \mathrm{WTK}(T_i, T_j) := \exp(-\lambda W_1(T_i, T_j))$ is not PSD. This leads to the conjecture that the metric may be influenced by characteristics of the time series. To address this issue, we may take several routes:

(a) We may compute the *empirical kernel* and thereby *enforce* the eigenvalue constraint. To do so, we calculate the kernel matrix $\mathcal{K}' := \mathcal{K} \cdot \mathcal{K}^\top$, where $\mathcal{K}$ is a matrix of dimension $k \times k$ whose entries follow Equation 3.12. To show that $\mathcal{K}'$ is PSD, let $\mathbf{y} := \mathcal{K}^\top \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^n$. We have $\mathbf{x}^\top \mathcal{K}\mathcal{K}^\top \mathbf{x} = \mathbf{x}^\top \mathcal{K}\mathbf{y} = \mathbf{y}^\top \mathbf{y} = \sum_{i=1}^{n} \mathbf{y}_i \geq 0$, hence $\mathcal{K}'$ is PSD.

(b) We may *regularise* $\mathcal{K}$ by subtracting all negative eigenvalues, leading to $\mathcal{K}' := \mathcal{K} - \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$, where i indexes the negative eigenvalues and $\mathbf{v}_i$ denotes their corresponding unit eigenvectors. This will set negative eigenvalues to zero, resulting in a positive definite matrix by construction.

(c) Following [283], we may *generalise* the Wasserstein distance to a "softmin" of all possible transportation plans. This guarantees that we get a PSD kernel.

(d) We may *sidestep* the eigenvalue constraint altogether by making use of algorithms capable of dealing with *indefinite* matrices [187].

In the following paragraphs, we will briefly explore and discuss these options.

ENFORCE Option (a) is computationally cheap, requiring only an additional matrix multiplication. However, the distance values between individual time series are changed, and we observed empirically that, compared to the other options, the predictive performance suffers to some extent.

REGULARISE Option (b) is computationally somewhat more expensive, as it necessitates a complete eigendecomposition of $\mathcal{K}$. Wu et al. [282] describe several transformations and illustrate that at least one of them, the spectrum `shift`, has minimal computational requirements and impacts classification performance only marginally.

GENERALISE Option (c) follows the "softmin" approach by Cuturi et al. [63], i.e.

$$\mathrm{SoftWTK}(\mathcal{M}_i, \mathcal{M}_j) := \sum_{\gamma \in \Gamma(\sigma,\mu)} \exp -\lambda \left(\int \mathrm{dist}(x,y)^p \mathrm{d}\gamma(x,y) \right)^{\frac{1}{p}}. \tag{3.13}$$

This is a kernel if $^{\text{dist}}/_{1+\text{dist}}$ is PSD. However, it involves the computation of D_{kl}'s permanent [59], which results in a computational bottleneck due to its super-exponentially scaling behaviour. This renders using SoftWTK infeasible; we therefore exclude it from our classification experiments.

SIDESTEP Option (d) is the option of our choice as we can use algorithms that can handle indefinite kernels directly without having to adjust $\mathcal{K}$. Such methods are a valuable approach for kernels that exhibit good predictive performance but do not satisfy positive definiteness (e.g. as shown by Vert [272]). These approaches utilise the so-called *Reproducing Kernel Kreĭn Space* (RKKS). In an RKKS, the kernel function does not have to be positive definite which leads to an indefinite kernel, i.e. they are neither negative definite nor positive definite. Previous research [100, 152, 164, 289] showed that it is possible to modify SVM classifiers so that they can work with indefinite kernels without resulting in decreased predictive performance.

Moreover, almost all data sets used in our experiments led to a PSD kernel matrix. Therefore, we call `WTK` a *kernel* and add that some data sets (for which a corresponding Kreĭn space exists) lack a corresponding Hilbert space. To ensure that our calculations are sound, we employ a Kreĭn SVM [158], which can handle both indefinite and positive definite matrices. In addition, we also explored Options (a) and (b). However, both did not lead to a significant performance gain with respect to `WTK`; in fact, the final test accuracies are virtually identical in all cases.

COMPLEXITY AND IMPLEMENTATION Computationally, `WTK` can be be split into the following steps: 1. Subsequence extraction, 2. Subsequence distance calculation, and 3. Wasserstein metric calculation. Letting k refer to the data set size, as introduced in Section 2.1.1, we have not more than $s := m - w + 1$ subsequences for each time series. Therefore, the time series length m dominates the extraction process, which takes the computational complexity of Step 1. to $\mathcal{O}(km)$. We share this preprocessing step with other subsequence-based methods, such as the original shapelet approach [287] or more recent approaches such as MPDIST [90]. Subsequently, we perform the following operations for each time series pair. For each time series pair, s^2 distance calculations are required. To compute the distance between *two* subsequences, w computations must be performed. Thus, the worst case complexity for Step 2 is $\mathcal{O}\big(s^2 w\big)$. Finally, to evaluate Equation 3.11 for a time series pair takes a complexity of $\mathcal{O}\big(s^3 \log s\big)$ for an $s \times s$ input matrix [6]. Asymptotically, the runtime of `WTK` can thus be expressed as $\mathcal{O}\big(k^2 m^3 \log m\big)$, as m is an upper bound on the number of subsequences of fixed length. Notably, this is only a worst-case approximation and several efforts (mostly based

Algorithm 3 Wasserstein Time Series Kernel

Input: Time series for training and testing $\mathcal{T}_{\text{train}}$, $\mathcal{T}_{\text{test}}$; subsequence length w; kernel weight factor λ

Output: $\mathcal{K}_{\text{train}}, \mathcal{K}_{\text{test}}$

// Extract subsequences
$\mathcal{M}_{\text{train}} \leftarrow \textsc{Subsequences}(\mathcal{T}_{\text{train}}, w)$
$\mathcal{M}_{\text{test}} \leftarrow \textsc{Subsequences}(\mathcal{T}_{\text{test}}, w)$
for $T_i \in \mathcal{T}_{\text{train}}$ **do**
    for $T_j \in \mathcal{T}_{\text{train}}$ **do**
        // Wasserstein distance calculation (train)
        $D_{ij}^{\text{train}} \leftarrow W_1\left(\mathcal{M}_i^{\text{train}}, \mathcal{M}_j^{\text{train}}\right)$
    end for
    for $T_k \in \mathcal{T}_{\text{test}}$ **do**
        // Wasserstein distance calculation (test)
        $D_{ik}^{\text{test}} \leftarrow W_1\left(\mathcal{M}_i^{\text{train}}, \mathcal{M}_k^{\text{test}}\right)$
    end for
end for
// Kernel matrix calculation
$\mathcal{K}_{\text{train}} \leftarrow \exp\left(-\lambda D^{\text{train}}\right)$
$\mathcal{K}_{\text{test}} \leftarrow \exp\left(-\lambda D^{\text{test}}\right)$
return $\mathcal{K}_{\text{train}}, \mathcal{K}_{\text{test}}$

on a Sinkhorn approximation) for obtaining *near linear-time* approximate solutions for the Wasserstein distance exist [17, 60]. In our experimental setup, however, the increased speed obtained using the Sinkhorn approximation was accompanied by a drop of predictive performance. Nevertheless, we did not extensively explore the hyperparameter space and only relied on a basic entropic regularisation scheme.

Algorithm 3 shows a pseudocode description of `WTK`. We require specifying a subsequence length $w \in \mathbb{N}_{>0}$ and a weight factor $\lambda \in \mathbb{R}$ for the similarity measure calculation in Equation 3.12. After extracting all subsequences, Equation 3.11 and 3.12 are used to create matrices $\mathcal{K}_{\text{train}}$ and $\mathcal{K}_{\text{test}}$, which can be used in classification algorithms such as SVMs. Our implementation used `Python 3.7` and `POT`, the *Python Optimal Transport* library [85]. Our code is publicly available[1].

[1] https://github.com/BorgwardtLab/WTK

3.2.3.3 Experimental Setup

In the following, we will investigate several aspects of our new kernel. First, we compare WTK to kernel approaches that are also subsequence-based to demonstrate that the naïve application of the $\mathcal{R}$-convolution framework can be meaningless. Moreover, we empirically show that using the 1-Wasserstein distance for comparing time series by means of their subsequences is well suited for TSC. This is followed by a comparison to DTW-1-NN in terms of a "Texas Sharpshooter" plot [16] to contrast *expected* and *actual* predictive performance measured on the training and test set, respectively. Such a plots enables us to demonstrate that WTK leads to consistently good predictions. Finally, we conclude our experiments with a large-scale performance assessment by comparing WTK to the state of the art of each data set.

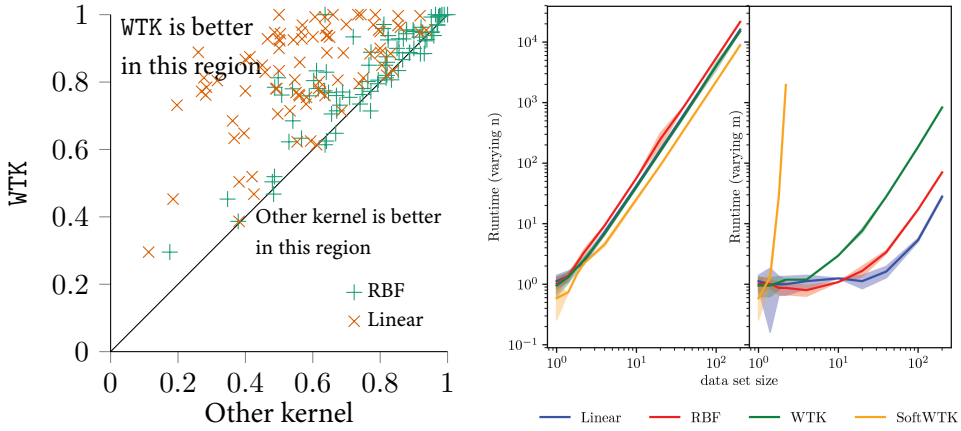
All experiments are performed on the *original* 85 data sets from the "UCR Time Series Archive" [66]. Each data set consists of predefined train/test splits of varying sizes and time series of multiple lengths; however, per data set, the length of all time series is always fixed. More details about the data sets can be found at https://www.timeseriesclassification.com. To evaluate the performance of WTK, different experiments are conducted. We are specifically interested in assessing accuracies obtained using WTK and contrast them with (a) other subsequence-based approaches, (b) baselines such as DTW, and (c) the state of the art in TSC.

Comparison partners We compare WTK to a residual network [105] architecture (ResNet), a fully convolutional network (FCN), and shapelet-based classifiers such as Learned Shapelets (LS) [95] and the Shapelet Transform (ST) [35]. Furthermore, we include established ensemble methods such as FLAT-COTE [11], Elastic Ensemble (EE) [154], and HIVE-COTE [155] in our comparison. Other algorithms include methods based on a symbolic aggregate approximation (SAX) of the time series [153] including DTW_F [128] and the SAX Vector Space Model (SAXVSM) [229] as well as classifiers that are dictionary-based such as the Bag of Symbolic Fourier Approximation Symbols (BOSS) [223] method. This approach combines DTW distances with SAX histograms. Furthermore, we consider a rotation forest (RotF) [211] containing 50 trees and a random forest [38] with 500 trees. Finally, other baselines include a Bayesian network (BN) and a 1-nearest neighbour classifier based on Euclidean distance (E-1NN).

Training and evaluation We evaluate classification accuracy on the predefined test sets and all parameters are determined using 5-fold cross validation on the training set. Parameters for the Kreĭn SVM classifier [158] are determined by evaluating validation performance on a parameter grid as follows:

- $\gamma \in \{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$ (for the RBF kernel)
- $\lambda \in \{10^{-4}, 10^{-3}, \ldots, 10\}$ (for `WTK`)
- $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ (for the SVM classifier)

Additionally, for methods that use subsequences, we vary their respective length w by assessing values of 10 %, 30 %, and 50 % of m. At first, we also investigated the classification performance of k-NN classifiers as they can use the distance matrix generated by Equation 3.11 directly. However, on average, these classifiers perform more than 3 % *worse*, so we neither discuss nor include them in the subsequent analysis.



(a) Comparison of accuracy of WTK and linear/RBF kernels for the "UCR Time Series Archive" data sets [66]. Ideally, all points are above the diagonal.

(b) Empirical runtime of four subsequence-based kernels when scaling the data set. Here, n denotes the number of samples and m the time series length. The y-axis shows the running time normalised with respect to the shortest-running method.

Figure 3.4: Comparison of WTK in terms of predictive accuracy and empirical runtime. © 2019 IEEE

3.2.3.4 Results

Other Kernels As a first experiment, we contrast our method with other kernels based on subsequences. For that, we train a standard linear kernel as well as an RBF kernel on subsequences of the same length. We showed in Section 3.2.3.1 that using a linear kernel simplifies into the comparison of time series means; hence, we expect to observe low accuracies in this scenario. In contrast, previous research showed [218] that the RBF kernel can lead to good predictive performance, but it has not been included in a large-scale study to the best of our knowledge. Due to its capability of capturing nonlinearities, we expect this approach to outperform the linear kernel. That being said, the RBF kernel only compares subsequences *independently*, while our method can compare *whole distributions* of subsequences, leading to a more expressive similarity measure. Figure 3.4 shows the results for all 85 UCR data sets. As expected, the linear kernel is outperformed by WTK in *all* cases. This empirically demonstrates the issues identified in the beginning: the $\mathcal{R}$-convolution framework can be meaningless if naïvely applied to time series subsequences. Contrasting our method with the RBF kernel, we observe that WTK outperforms it on 73 out of 85 data sets. Even so, as illustrated in the plot, the accuracy deviation for the points below the diagonal is insignificant, and the mean accuracy difference for these data sets is only ≈2.2 %. This shows that our method's competitive predictive performance is not due to the consideration of subsequences in itself, but by taking into consideration the distribution of their similarities.

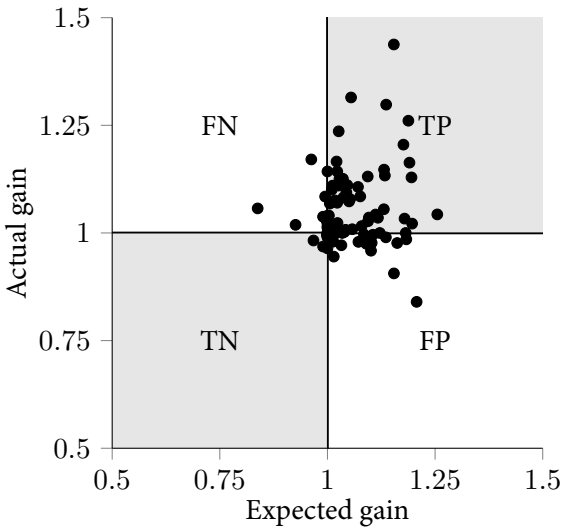


Figure 3.5: "Texas Sharpshooter" plot, comparing *expected* gains to *actual* gains, relative to DTW-1-NN. © 2019 IEEE

To conclude this experiment, a brief empirical runtime analysis is performed. Figure 3.4b confirms that computing the Wasserstein distance is not the decisive factor when it comes to runtime performance. The linear and RBF kernel, as well as WTK, all use subsequence differences and show the same asymptotic runtime behaviour. Moreover, we can observe SoftWTK's super-exponentially scaling behaviour, which is due to the aforementioned computation of the permanent.

Comparison with DTW-1-NN Dau et al. [66] pointed out that a 1-nearest neighbour classifier trained on DTW distances constitutes a strong baseline in time series classification. We compare `WTK` to DTW-1-NN, in terms of a "Texas Sharpshooter" plot [16]. It visualises the "expected" gain as measured on the training set on the x-axis and the *actual* gain as measured on the test set on the y-axis. For both methods, we used 5-fold cross-validation to identify the best parameters and expected gains.

The results of this analysis for all data sets are depicted in Figure 3.5. Almost all points are in the TN or TP quadrants, suggesting our method is either consistently outperformed by DTW-1-NN (TN) or consistently outperforms DTW-1-NN (TP). Some points are in the remaining two quadrants. False negative (FN) points are a "happy surprise" because we expected our method to perform worse than it does. The most problematic quadrant is the false positive (FP) region; however, it only contains a few points and the differences in accuracy are comparatively minor. Overall, the sharpshooter plot yields evidence that, in terms

of predictive performance, WTK is superior to DTW-1-NN, as the TP quadrant contains the majority of points.

Comparison with the state of the art In our final analysis, we compare WTK with the SOTA in TSC. For this, we compiled the accuracies of *all* methods that were published in the "UCR Time Series Classification Repository" [66] at the time of this study. Additionally, we collected classification performances of two high-performing neural network approaches [278] whose classification performances are provided in a review by Fawaz et al. [79]. Overall, we collected results from 40 methods; however, the results of the neural network approaches were incomplete. For each data set, we selected the method that performed best on the published test set. We therefore compare WTK to the 40 top-performing methods, which ensures the most comprehensive and honest testing scenario. WTK exceeds prediction performance of *all* state-of-the-art methods on the following six data sets DistalPhalanxTW, DistalPhalanxOutlineAgeGroup, MiddlePhalanxOutlineAgeGroup, ECG5000, Earthquakes, and FordB. Furthermore, our method reaches the same accuracy as the state of the art on BeetleFly, ECGFiveDays, Coffee, Plane, Trace, and ShapeletSim, leading to a total of 12 data sets for which WTK is at least as good as the respective SOTA method. That being said, the TSC community considers some of these data sets to be solved.

Performance Details There are numerous other data sets on which our method's performance is close to the state of the art. Table 3.1 provides a more detailed analysis of these accuracy differences. We contrast WTK with HIVE-COTE, the best-performing approach, and KEMD, a competitor that is conceptually similar as it is also OT-based. Each row in the first column defines a range of accuracy differences (with respect to the SOTA performance) on which the remaining three columns are conditioned. Thus, these columns show the percentage of data sets for which the respective method fulfils the condition of the first column. While overall, HIVE-COTE (an ensemble method) outperforms our approach, for almost 45 % of all data sets, WTK's performance difference does not exceed 5 %. In contrast, KEMD's performance seems to be relatively erratic with favourable performances on a small number of data sets, while being more decisively outperformed on most of them.

Statistical Analysis To underline the effectiveness of our proposed method, we show a *critical difference* (CD) *plot* [69] in Figure 3.6 that contrasts WTK with a variety of competing methods (due to typographical reasons, we refrain from displaying a comparison with *all* 40 methods, but the relative ranking remains the same). If a bold horizontal line con-

Table 3.1: The first column defines a condition over the absolute difference (Δ) in mean accuracy compared to the best performing method (per data set). The remaining columns show the fraction of data sets for which the respective condition is fulfilled. Due to rounding, columns do not sum to 100 %. © 2019 IEEE

Δ	WTK	HIVE-COTE	KEMD
$\Delta \geq 0$	14.1 %	36.5 %	4.7 %
$0\% > \Delta \geq -5\%$	44.7 %	34.1 %	15.3 %
$-5\% > \Delta \geq -10\%$	24.7 %	18.8 %	7.1 %
$-10\% > \Delta \geq -15\%$	8.2 %	1.2 %	16.5 %
$-15\% > \Delta \geq -20\%$	4.7 %	7.1 %	9.4 %
$-20\% > \Delta$	3.5 %	2.4 %	47.1 %

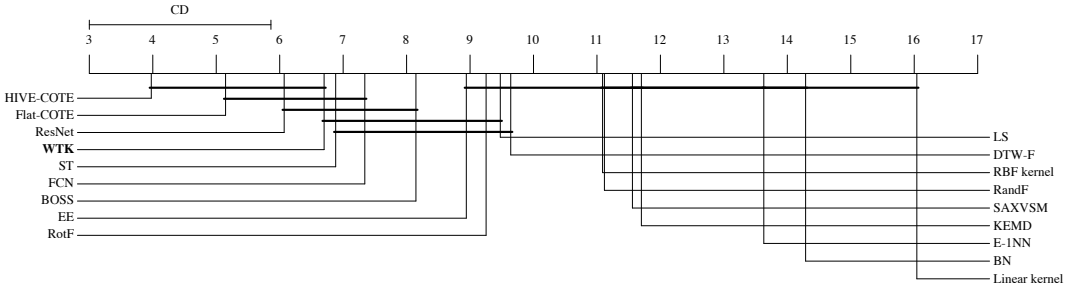


Figure 3.6: Comparison of WTK (in bold) with top-performing competitor methods by means of a critical difference (CD) plot. We observe that the performance of our method is not statistically significantly different to the state of the art. © 2019 IEEE

nects two methods, their performances are not statistically significantly different from each other. At a significance threshold of $\alpha = 0.05$, we observe that there is no statistically significant difference between our method and the best-performing classifiers. Since the top-performing methods are either heavily-parametrised (e.g. deep neural networks) or *ensembles* constructed from more than 30 other methods, the plot underlines good generalisation performance of WTK.

Comparison with selected methods Figure 3.7 illustrates accuracy differences for selected methods in more detail. For this juxtaposition, we chose the best method (HIVE-COTE), the best neural network (ResNet), and KEMD, which is conceptually similar. Each scatter plot shows the respective accuracies obtained on a given data set. Our method's performance follows ResNet's accuracy values closely, as most points are situated near the diagonal. By contrast, we clearly outperform KEMD on almost all data sets.

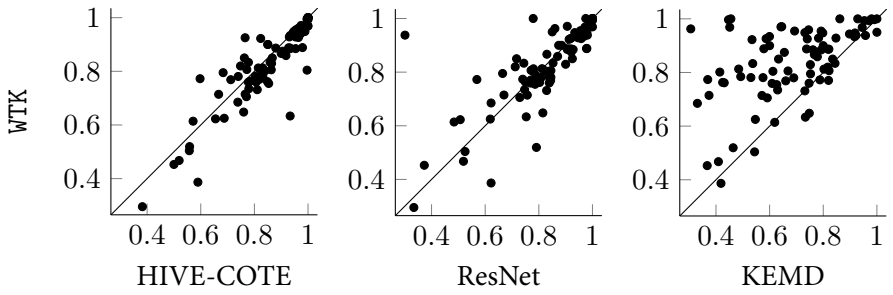


Figure 3.7: Comparison of WTK with two well-performing methods. Each dot represents the performance values (accuracy) of two approaches (axes) on one data set. Axes only show accuracies between 0.4 and 1.0 since all values are within this range. © 2019 IEEE

3.2.4 Conclusion

In this section of the thesis, we showed theoretically and experimentally that $\mathcal{R}$-convolution kernels cannot be naïvely adapted to time series, as they can degenerate to a simple comparison time series means. This motivated the development of WTK, a new subsequence-based kernel that utilises concepts from optimal transport theory. More precisely, it leverages the Wasserstein distance to compare distributions of subsequences that serve as time series representations. We investigated the expressiveness of this similarity measure in the time series classification setup by performing an extensive evaluation on the "UCR Time Series Archive" data sets. Our performance analyses indicate that the proposed method can outperform the state of the art in time series classification and exhibits good generalisation properties. This shapelet-based method broadens the scope of this thesis by shifting from a pattern mining perspective that is mainly concerned with statistical associations of shapelets to the time series classification task utilising shapelets.

3.3 Predicting Stress-Induced Myocardial Ischaemia from ECG-Recordings

In the preceding sections, we developed a domain-independent TSC method of general applicability by identifying methodological shortcomings in other approaches. In this section, we will focus on a time series classification problem that arises in a clinical environment. In general, time series classification plays a vital role in healthcare settings in which data is frequently measured. This includes the continuous monitoring of vital parameters as detailed before and extends to electrocardiograms (ECGs), one of the central data types in cardiology [92]. The ECG provides a cardiologist with detailed information about the heart's electrical activity from which many cardiac pathologies can be inferred. The potential diagnostic and prognostic value of machine learning in cardiology range from aiding clinical decision making (e.g. determining whether a specific test should be performed or not) to workflow optimisation to the enhancement of diagnosis and risk stratification [159]. This part of the thesis is based on a manuscript that is in preparation and assesses the clinical value of a collaborative deep learning system for the detection of exercise-induced myocardial ischaemia (EIMI). It is structured as follows: First, we will highlight the relevance of predicting exercise-induced myocardial ischaemia and the clinical protocol to determine this pathology. Second, a description of the development of our prediction system is provided including details of the data set, the machine learning approach, and the evaluation scenarios. We then present results on a held-out test set focusing on clinically relevant performance metrics, followed by an interpretability analysis.

9.

3.3.1 Introduction

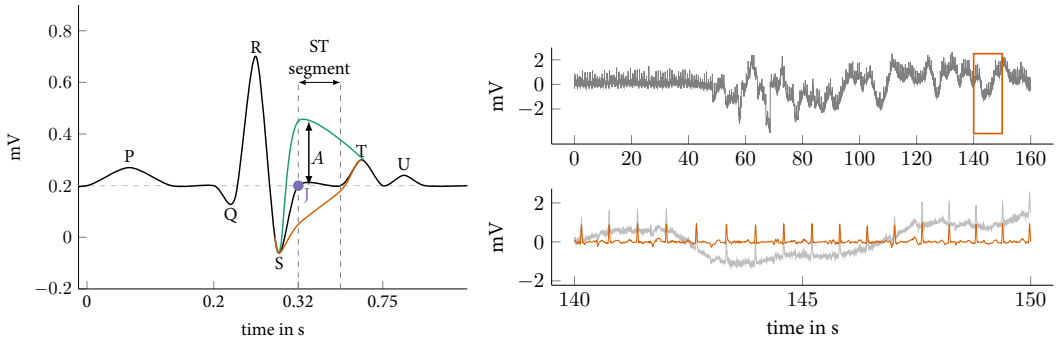
Ischaemic heart disease, i.e. the undersupply of oxygen to the heart, is the leading cause of years of life lost (YLL) worldwide [120]. This means that over 10 % of all lives that were lost due to premature death are linked to cardiovascular complications. High mortality and morbidity rates paired with the availability of cost-efficient prevention measures underline the importance of early risk-stratification of patients with suspected myocardial ischaemia. The practical utility of current screening techniques, however, is limited by either unfavourable diagnostic accuracy, as in the case of exercise electrocardiography stress testing, or by its invasive nature and high costs, as in the case of myocardial perfusion imaging (MPI) [196]. Ideally, a risk-stratification tool should be 1) non-invasive, 2) easily accessible to patients at risk, 3) cost-efficient, and 4) of high clinical value (which can take several forms).

Traditionally, the automated prediction of cardiac events employed methods that rely on the quantification of ECG changes such as ST-segment changes, T-wave abnormalities, or other anomalies in the ECG (see Figure 3.8a for a schematic illustration of an ECG). Methods based on the manual extraction of wave-specific features necessitate not only involved and error-prone preprocessing steps such as ECG delineation, but are also limited in clinical accuracy. Furthermore, when it comes to very long time series and larger sample sizes, the straightforward application of the methods we introduced in Chapter 2 and Section 3.2 fail due to scalability issues. Circumventing the problem of scalability and the elaborate definition and computation of ECG-specific features while increasing clinical accuracy was recently achieved by employing deep learning. Successes can be found in tasks such as diagnosis prediction, rhythm and form recognition, and cardiovascular death prediction [42, 101, 181, 203, 248].

The system we will develop in the course of this and the following sections predicts EIMI, an early indicator of future cardiac complications, from easy-to-access static and ECG data only. We systematically assess the value of data preprocessing, lead-selection, and the relevance of multi-task learning for predictive performance. The clinical utility of the developed system, which we refer to as Neural Ischaemia Prediction (`NIP`), lies in its ability to reduce the false positive rate (FPR) of a physician from 0.90 to 0.75 while maintaining high sensitivity (0.95). This has the implication that if we relied on our system to decide whether a patient should receive an MPI, the number of patients that unnecessarily undergo this procedure would drop by 15 percentage points. In a subgroup of patients with a previous history of CAD that were able to complete the exercise stress test without pharmacological support, a combination of `NIP` and the physician's judgement (`NIP+`) reduced FPR *and* false negative rate (FNR) by 0.21 and 0.11 percentage points, respectively. Moreover, we provide an interpretability study that sheds light on the inner workings of the model, helping the cardiologist to understand how different clinical features and ECG segments contribute to `NIP`'s predicted risk score.

3.3.1.1 The ECG and Exercise Stress Testing

The electrocardiogram (ECG) is the main tool to retrieve information about the electrical activity of the heart. Twelve (sometimes up to 18) electrodes, applied to different parts of the body, measure electrical changes of the skin that arise from depolarisation and repolarisation of the cardiac muscle. The different electrodes or leads are, depending on their position, labelled as `I`, `II`, `III`, `aVL`, `aVR`, and `aVF` (**a**dvanced **V**ector **L**eft/**R**ight/**F**oot) for the limb leads, and `V1` through `V6` for chest leads. Figure 3.8a sketches a "typical" QRS complex as it is visible in heart beats recorded at lead `II`. The P-wave reflects atrial depolarisation, resulting



(a) Schematic illustration of a QRS complex (lead `II`). The *ST-Amplitude A* is the vertical distance of the *ST-segment* to the isoelectric line. The ST-segment can be elevated (green) or depressed (orange). Furthermore, the direction of the ST-segment slope (upsloping, downsloping) plays an important role in ischaemia prediction.

(b) The upper plot shows 160 seconds of a raw ECG signal from lead `II`. Starting after approx. 50 s, the signal is heavily influenced by noise from patient movements. The lower plot illustrates a window of 10 seconds of the respective signal. The orange curve depicts the results of a preprocessing pipeline. The original signal is visible in light grey.

Figure 3.8: Schematic illustration of a lead `II` ECG of a single heart beat (left) and an example of a lead `II` ECG signal (right).

in atrial activation and is followed by the "QRS complex". This complex represents how an electric stimulus spreads through the ventricles of the heart [92]. In the ECG, this is visible as the characteristic Q (first negative deflection), R (first positive deflection), and S wave (first negative deflection after R wave). Ventricular recovery is represented by ST-segment and T wave (repolarisation). The junction or J-point is where QRS complex and ST-T wave meet and the point at which the amplitude (A) of the ST-segment is measured. In exercise stress testing, ST-amplitude is often measured at 40 ms, 60 ms, or 80 ms after the J-point. Finally, the U wave is a low-amplitude wave whose electrophysiologic basis is not certain [174] and which is sometimes referred to as "the enigmatic sixth wave of the ECG" [210]. Figure 3.8b depicts a real lead-`II` ECG from exercise stress testing. Several sources of noise such as baseline wander, powerline interference, electrode motion artefacts, and electromyographic noise are prevalent in almost all stress test ECGs. This necessitates an appropriate signal preprocessing pipeline preceding any automated analysis [9].

Exercise Stress Testing In order to diagnose EIMI, an experimental setup with bicycle ergometry is employed. In an initial *pre-stress* phase, the patient exercises on the bicycle with zero resistance. The resistance is then increased iteratively aiming to make the patient reaching a predefined, patient-specific heart rate. Once this heart rate is reached, the resistance is reduced and the patient enters a *recovery* phase. During all three phases (*pre*, *stress*,

and *recovery*), the cardiologists monitors the 12 leads of the ECG, including summary heart beats of lead `II`, heart rate, and blood pressure. The physician is also in constant exchange with the patient about their well-being. Once a patient reaches their peak heart rate, a myocardial perfusion scan (MPS) is performed by injecting a radioactive tracer intravenously to record how the tracer perfuses through the ventricles of the heart via single-photon emission computerized tomography (SPECT) imaging. Based on a second MPS, taken at rest, the cardiologist will determine to which extent the ejection rate has changed and whether myocardial ischaemia was induced. Not all patients are able to reach their target heart rate or to exercise at all. These patients are switched to a pharmacological stress testing protocol. In both cases, the subjects will receive a pharmacological substance that either mediates coronary artery vasodilation (adenosine) or increases cardiac output and heart rate (dobutamine). Therefore, each stress test falls into one of four categories:

- Regular exercise stress test
- Fully pharmacologically-induced stress test using adenosine
- Fully pharmacologically-induced stress test using dobutamine
- Combined exercise and pharmacological stress test

The last situation describes a scenario in which a patient starts with exercising but requires pharmacological support to reach their target heart rate. For subsequent subcohort analyses, we combined patients for which a full pharmacological protocol was followed into a single group.

3.3.1.2 Machine Learning for Cardiology

The automated detection of myocardial ischaemia and infarction has a long history and was recently reviewed by Ansari et al. [9]. In the following paragraphs, we focus on recent promising contributions that make use of deep learning to increase classification performance in several tasks relevant for the field of cardiology.

Risk Stratification Myers et al. [181] developed a risk-stratification tool for patients that suffered from acute coronary syndrome (ACS). Their proposed recurrent neural network received transformed descriptors of ST-segment slope and amplitude as well as static patient features as input signal. This way, the authors increase the prediction performance of cardiovascular death (CVD) within one year after ACS in patients without ST-segment elevation. Shanmugam et al. [231] study the impact of including the whole ECG signal on risk

stratification performance on the same prediction task. More specifically, the authors utilise the multiple instance learning framework [168] and single-lead ECG signals to predict CVD within 30, 60, 90, and 365 days of hospital admission. Over all time horizons, their two-layered convolutional neural network (CNN) outperforms other approaches in terms of area under the receiver operating characteristic (AUROC) and odds ratio.

Diagnosis and Arrhythmia Prediction In addition to risk stratification, deep learning has been successfully utilised for diagnosis prediction, and form and rhythm detection of ECG signals. The latter was approached by Hannun et al. [101], where a deep learning system was trained to detect different rhythm classes (e.g. atrial fibrillation, atrioventricular block, or ventricular tachycardia). Trained on only a single-lead ECG, the system performed better than the average cardiologist. In a similar experimental setup but with access to 12 leads, the neural network by Ribeiro et al. [203] predicted ECG abnormalities in addition to diagnostic statements such as right bundle branch blocks. Their approach performed better than 4$^{\text{th}}$ year cardiology residents and has significantly fewer parameters than the first hallmark study by Hannun et al. More recently, Strodthoff et al. [248] extracted a hierarchical set of 44 diagnoses from over 15 000 ECG records (12-lead, 10 s) and assessed the performance of a variety of deep learning architectures in several multi-label classification tasks. Diagnoses include myocardial infarction, conduction disturbances, and ST-T changes. In the same work, the authors predicted 19 form statements (e.g. non-specific ST changes, low amplitude T-waves, or abnormal QRS complexes) and 12 ECG rhythm labels (e.g. sinus bradycardia or atrial flutter).

Many of these studies were possible due to the automatic extraction of labels (e.g. from ECG reports), which makes data generation very efficient. Furthermore, many rhythm and form classes manifest themselves in permanent changes of the ECG. Determining the presence of exercise-induced myocardial ischaemia, however, requires an elaborate stress test during which a potential ECG signal may develop transiently. Additionally, to determine the ground truth label, SPECT images need to be interpreted and a resulting diagnosis may be refined using fractional flow reserve measurements and coronary angiography. This is what differentiates EIMI prediction from earlier work on cardiovascular event detection; the physiological signal of this complex phenotype develops throughout the experiment and it may even vary between different subcohorts.

3.3.2 Cardiologist-Level Ischaemia Prediction with Deep Learning

In this section we detail the development of our system for EIMI prediction, which we refer to as Neural Ischaemia Prediction (`NIP`). We structured the section as as follows. First, we

Table 3.2: Static clinical features used for EIMI prediction.

Name	Type	Description
Age	Numerical	Age of patient in years
Sex	Binary	Biological sex of patient[2]
Height	Numerical	Height of patient in cm
Weight	Numerical	Weight of patient in kg
Resting HR	Numerical	Heart rate at rest. Measured in beats per minute (BPM)
Resting Sys. BP	Numerical	Systolic blood pressure at rest. Measured in millimetre of mercury (mmHg).
Resting Dias. BP	Numerical	Diastolic blood pressure at rest. Measured in millimetre of mercury (mmHg).
Known CAD	Binary	Absence/presence of previous coronary artery disease

describe the process of acquiring training data and labels during the exercise stress test. We will also establish a "human baseline" that will serve as point of reference when assessing the performance of our system. This is followed by a more detailed description of the employed multi-task learning approach that we take. Before detailing other feasible approaches for ischaemia prediction (i.e. our experimental comparison partners), a detailed overview of the experimental setup is provided. This includes an overview of different signal preprocessing steps, a description of `NIP`'s architecture, and elucidations on the process of lead and (hyper)parameter selection.

3.3.2.1 Data Set

We conducted our experiments on 12-lead ECGs from 3522 patients from the `BASEL-VIII` study (ClinicalTrials.gov registry, number NCT01838148). Around one third of all ECGs were downsampled from 1000 Hz to 500 Hz, leading to time series with a median length of 476 589 measurements, or 15 min, respectively. In addition to ECG signals, we further included static clinical features such as age, sex, height, blood pressure, heart rate, and the presence/absence of previous CAD in our assessment (see Table 3.2 for a complete list).

Data Acquisition Figure 3.9 illustrates the data acquisition process as part of the ECG stress test. All 3522 patients underwent a standard [271] rest/stress MPI-SPECT/CT protocol

[2] We use the term "biological sex" as used in many epidemiological studies. We understand, however, that sex involves multiple social and biological factors [126] and can be different from the sex assigned at birth. Moreover, from a clinical perspective it is important to differentiate this way as the prevalences between both groups vary significantly.

using either one (^{99m}Tc sestamibi for both rest and stress) or two (^{201}TI for rest, ^{99m}Tc sestamibi for stress) radioagents. As shown in the figure, only eight ECG leads were measured directly, the others (`III`, `aVL`, `aVR`, and `aVF`) were computed according to Einthoven's triangle and Goldberger's equations [174]. If a patient was not able to reach their target heart rate (28.5 %), a pharmacological protocol with either adenosine or dobutamine was initiated. Individuals for which stress test by bicycle ergometry was not an option (17.9 %), either due to left bundle branch block (LBBB), presence of a pacemaker, or the inability to exercise, were put on the pharmacological protocol from the start. To compare the algorithmic approach to expert judgment, the treating cardiologist overlooking the stress test performed a clinical assessment before (pre-stress) and after (post-stress) the examination. Considering all available medical information such as cardiac history, relevant symptoms, baseline ECG, and more, the cardiologist indicated the probability of CAD/EIMI on a visual analog scale (VAS) from 0 % to 100 % [147, 243, 256, 277]. An expert team composed of a nuclear medicine physician and a cardiologist, assessed myocardial perfusion scans (MPS) on a semi-quantitative score using a 17 segment bull's eye scheme with a 5-point scale. For each segment, a score between 0 (normal tracer uptake) and 4 (no tracer uptake) is assigned and the sum over all segments summarises myocardial perfusion at rest (MPSSRS) and during stress (MPSSSS).

3.3.2.2 Multi-Task Learning & Auxiliary Tasks

Multi-task Learning (MTL) [47] is a technique to introduce inductive biases into neural networks to increase generalisation performance and decrease the risk of overfitting [215]. Its effectiveness has been demonstrated in the field of arrhythmia detection [121, 264] and prediction of other clinical phenotypes [46, 103, 213]. Given a primary task, related auxiliary tasks are introduced that will serve as mutual sources of inductive bias for each other. In other words, learning to solve multiple related tasks simultaneously "nudges" the network to learn representations that are useful across all tasks and hence across the domain at hand. In contrast to self-supervised learning, where tasks are derived from the input signal itself, MTL is inherently supervised in the classical way: label acquisition is a manual and elaborate process, typically involving domain experts, such as cardiologists in our case.

Formally, let $g_{\tau_\alpha}(f_\theta(X))$ be a classifier for the primary task α, where f_θ is a neural network consisting of *shared* layers parametrised by θ and g_{τ_α} represents the *task-specific* layers with parameters τ_α. Furthermore, X denotes the input data and Y_α the labels of task α. We can introduce an auxiliary task β by re-using the latent representation $f_\theta(X)$ to predict the task-specific label Y_β using another set of *task-specific layers*: $g_{\tau_\beta}(f_\theta(X))$. For each task $\gamma \in \Gamma$,

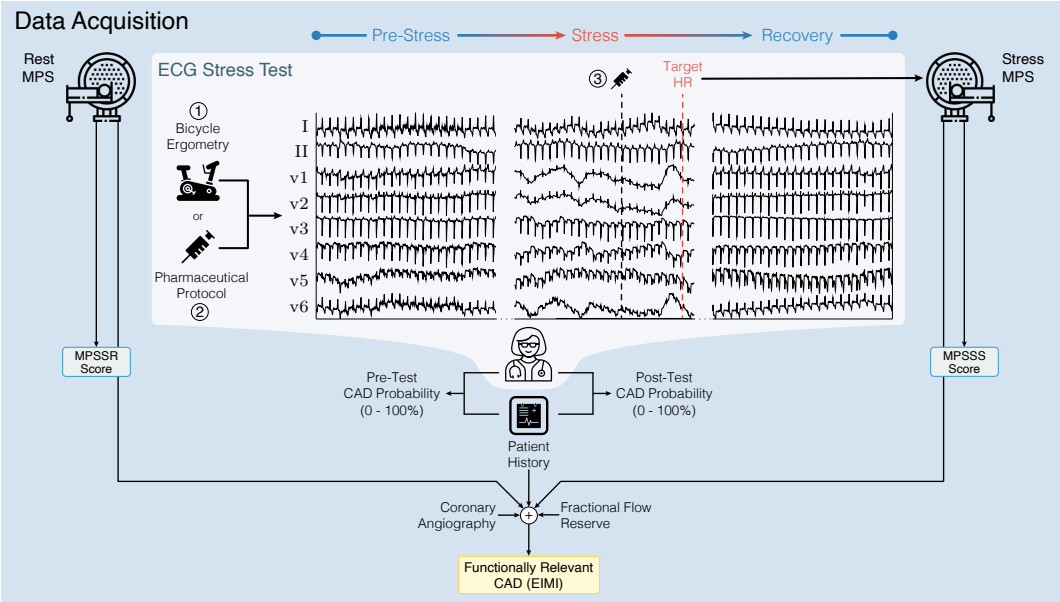


Figure 3.9: Overview of the data acquisition workflow. The three main subgroups of the exercise stress test are highlighted: (1) patients that complete the exercise stress test on the bicycle, (2) patients that are not able to exercise on the bicycle and for whom a pharmaceutical protocol is used, and (3) patients who start on the bicycle but need pharmacological support to reach their target heart rate. Both at rest and at the point where the patient reaches their target heart rate, a myocardial perfusion scan is performed. Both scans enable the derivation of two relevant scores, namely MPSSRS and MPSSSS. The treating cardiologist estimates the probability of a functionally relevant CAD before and after the stress test (Pre/Post-Test CAD Probability). The final binary label of presence of functionally relevant CAD is adjudicated by taking the stress test results and additional relevant clinical parameters into account.

we define a task-specific loss function $\mathcal{L}$ that measures the prediction error of g to arrive at the overall loss term to be minimised:

$$\mathcal{L} = \sum_{\gamma \in \Gamma} \big(\lambda_\gamma \mathcal{L}_\gamma(g_{\tau_\gamma}(f_\theta(X)), Y_\gamma)\big) + \mathcal{L}_\alpha(g_{\tau_\alpha}(f_\theta(X)), Y_\alpha) \tag{3.14}$$

In the preceding equation the magnitude of each auxiliary loss is regularised by λ. This instance of MTL is called *hard parameter sharing*, as exactly *one* set of parameters (θ) is shared between all tasks. We considered the following three auxiliary tasks whose respective regularisation parameters are denoted as, λ_{Stress}, λ_{MPSSRS}, and λ_{MPSSSS}. We used binary cross entropy ($\mathcal{L}_{\text{BCE}}$) as loss function to learn the main task.

Stress Type The "Stress Type" variable refers to the way stress was induced in the patient. More specifically, the network is challenged to predict whether they were able to exercise throughout the whole examination, whether adenosine or dobutamine was used to induce stress pharmacologically, or if a hybrid approach was taken. 1417 patients (53.5 %) in the development data set (see Section 3.3.2.3) underwent complete exercise stress testing, 416 patients (15.7 %) adenosine-induced testing, 59 patients (2.2 %) dobutamine-induced testing, and 756 (28.5 %) combined stress testing. We used a cross-entropy loss ($\mathcal{L}_{\text{CE}}$) to learn this task.

MPS Summed Rest/Stress Score As touched upon in Section 3.3.2.1, a commonly used semi-quantitative way of judging the effectiveness of the heart is to determine the uptake of the radioactive tracer in the left ventricle by assessing the MPS images [182]. Using a scale from 0 to 4, a cardiologist and a nuclear medicine physician provided scores for 17 segments of the muscle and the cavity of the left ventricle. Each segment gets assigned a score of normal (0), mildly reduced (1), moderately reduced (2), severely reduced (3), or no tracer uptake (4). The overall myocardial perfusion scan (MPS) score is computed by summing the individual scores of each segment. This evaluation is done at rest (MPSSRS) and under maximal stress (MPSSSS). To learn these tasks, we utilised the root mean squared error ($\mathcal{L}_{\text{RMSE}}$).

3.3.2.3 Experimental Setup

We split the data set 3:1 into development and held-out test set containing 2648 and 874 patients, respectively. During the development of the model, we had no access to the held-out test set. Access was provided, once we fixed all model parameters. The development set

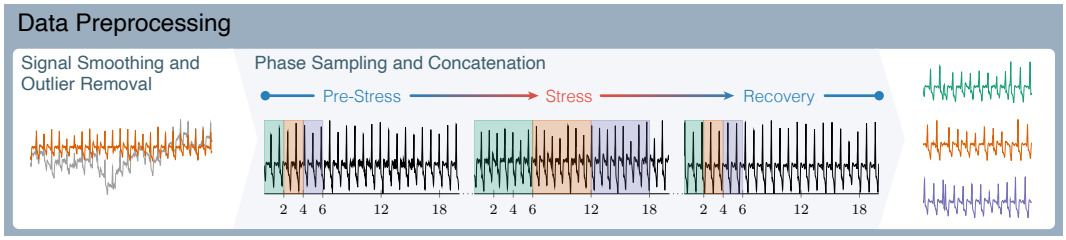


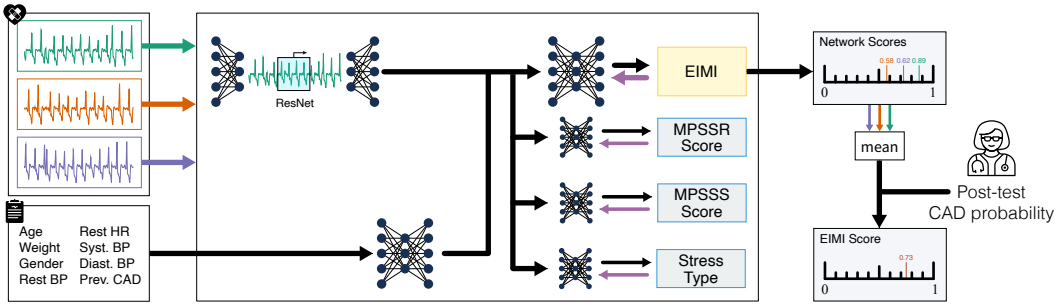
Figure 3.10: Illustration of data preprocessing and sampling steps. Time series that serve as input to the neural network are constructed by concatenating short subsequences from different phases of the stress test.

was further divided into 5 stratified splits of training, validation, and test set, where the latter makes up 10 % of the development set. The ratio of training to validation set size is 4:1.

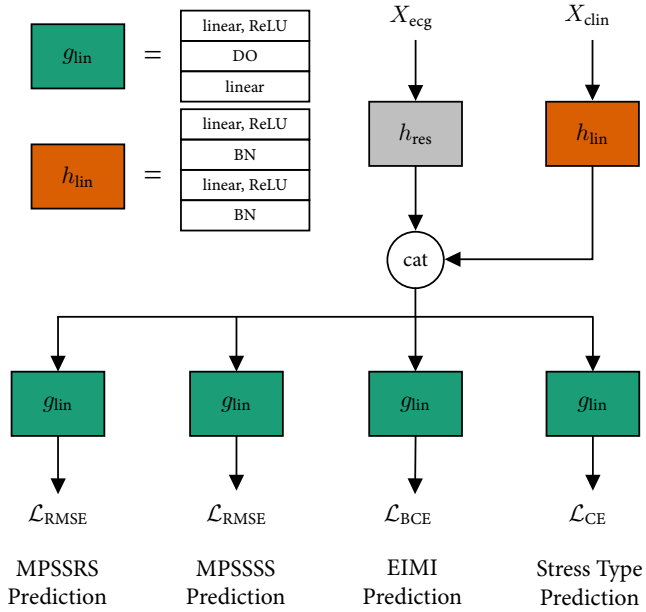
ECG Input Generation After applying a set of signal preprocessing steps (see below), we construct the input time series by concatenating subsequences from different phases of the examination as illustrated in Figure 3.10. For this, we sample two seconds from the beginning of the examination, six seconds from the last two minutes of the stress phase, and two seconds from the last three minutes of the recovery phase, and merge them into a single time series. This sequence, which we refer to as "2-6-2", was constructed up to twenty times per patient using a tumbling window for each experimental phase (indicated by differently coloured windows in Figure 3.10). This way the neural network receives an input that represents the complete examination in one sample with a focus on the stress phase.

Signal Preprocessing Schemes As visualised in Figure 3.8b, ECG signals from exercise stress testing are subject to high levels of noise from various sources. To assess the influence of noise on classification performance, we consider the following preprocessing schemes: 1. no preprocessing, 2. minimal preprocessing with a high-pass Butterworth filter of order five, and a cutoff frequency of 0.5 Hz followed by moving average smoothing, and 3. a thorough preprocessing pipeline consisting of a wider bandpass filter (0.05 Hz to 150 Hz), moving-median subtraction to remove baseline wandering, a Savitzky–Golay [221] filter for smoothing, and winsorising to deal with spurious outliers.

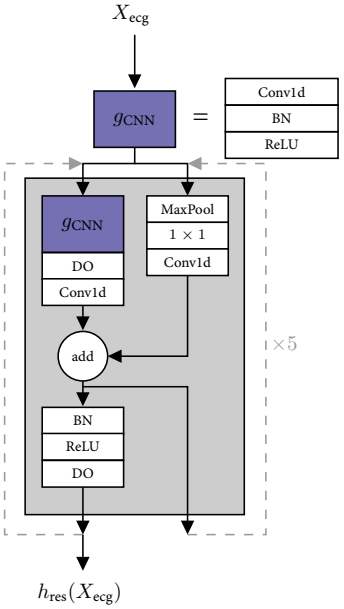
Architecture Figure 3.11a provides a high level overview of the multi-task learning setup. For each patient, up to 20 "2-6-2" sequences are constructed (three are shown in green, orange, and purple). Each sequence represents *one* training sample and is combined with the static features of the respective patient. This means, that the number of EIMI predictions per patient is the same as the number of "2-6-2" sequences that were generated



(a) Architectural overview of our EIMI prediction system. Individual input time series are constructed as visualised in Figure 3.10 and then fed into a residual neural network. In parallel the patient's static data are processed by a 2-layer feedforward network. Four subnetworks are trained on different tasks and predictions of the main task are aggregated and combined with the treating cardiologist's judgement.



(b) Composition of the multi-task architecture. Each task obtains its own loss function $\mathcal{L}$, and `cat` denotes the concatenation of the embeddings of the ECG signal X_{ecg} and the clinical data X_{clin}.



(c) Residual neural network h_{res} that takes the ECG signal as input.

Figure 3.11: (a): High-level overview of employed multi-task architecture and combination with physician judgement. (b) & (c): Detailed illustration of individual task-specific subnetworks.

Table 3.3: Architectural details of the used neural network. Convolutional layers are written as [input dimension, output dimension, kernel size, stride]$_{\text{Conv}}$, linear layers as [input dimension, output dimension]$_{\text{Lin}}$. BN: Batch norm, ReLU: Rectified Linear Unit, DO: Dropout. Max pooling is written as MP(kernel size, stride). "add" denotes the addition of the output of the $\text{MP}_{1\times1}$ layer and the preceding convolutional layer as shown in Figure 3.11c.

	Task	Layer Name	Parameters
	N/A	$\text{MP}_{1\times1}$	$[\text{MP}(4,4), [64, 128, 1, 1]_{\text{Conv}}]$
	N/A	BR	[BN, ReLU]
	N/A	BRD	[BR, DO(0.8)]
	N/A	$\text{Conv}_{\text{init}}$	$[[1, 64, 20, 1]_{\text{Conv}}$, BR]
	N/A	Res_1	$[[64, 128, 20, 1]_{\text{Conv}}$, BRD, $[128, 128, 20, 4]_{\text{Conv}}$, add, BRD]
	N/A	Res_2	$[[128, 196, 20, 1]_{\text{Conv}}$, BRD, $[196, 196, 20, 4]_{\text{Conv}}$, add, BRD]
	N/A	Res_3	$[[196, 256, 20, 1]_{\text{Conv}}$, BRD, $[256, 256, 20, 4]_{\text{Conv}}$, add, BRD]
	N/A	Res_4	$[[320, 320, 20, 1]_{\text{Conv}}$, BRD, $[320, 320, 20, 5]_{\text{Conv}}$, add, BRD]
	N/A	Res_5	$[[320, 160, 20, 1]_{\text{Conv}}$, BRD, $[160, 160, 20, 4]_{\text{Conv}}$, add, BRD]
	Embedding ECG	h_{res}	$[\text{Conv}_{\text{init}}, \text{Res}_1, \text{Res}_2, \text{Res}_3, \text{Res}_4, \text{Res}_5$, BR]
	EIMI Prediction	g_{lin}	$[[672, 32]_{\text{Lin}}$, ReLU, DO(0.5), $[32, 1]_{\text{Lin}}]$
	MPSSRS Prediction	g_{lin}	$[[672, 32]_{\text{Lin}}$, ReLU, DO(0.4), $[32, 1]_{\text{Lin}}]$
	MPSSSS Prediction	g_{lin}	$[[672, 32]_{\text{Lin}}$, ReLU, DO(0.4), $[32, 1]_{\text{Lin}}]$
	Stress Type Prediction	g_{lin}	$[[672, 32]_{\text{Lin}}$, ReLU, DO(0.4), $[32, 5]_{\text{Lin}}]$
	Embedding Clinical Features	h_{lin}	$[[8, 16]_{\text{Lin}}$, ReLU, BN, $[16, 32]_{\text{Lin}}$, ReLU, BN, DO(0.5)]

for that specific patient. In parallel, the patient's static data is embedded by a neural network whose output is concatenated to the residual network's output. The resulting representation serves as input to four subnetworks, each of which is responsible for the prediction of one of the four tasks, respectively, as described in Section 3.3.2.2. All EIMI predictions of a patient are aggregated to obtain the `NIP` risk score by taking their mean. To make use of the cardiologist's expertise if/when available, we combine their post-stress VAS score with `NIP`'s prediction by training a logistic regression on the EIMI task. We refer to the output of this logistic regression as `NIP+`. Figure 3.11 details the individual components of our neural network architecture. Following the notation in Section 3.3.2.2, all tasks share the representation generated by concatenating the outputs of h_{res} and h_{lin}, i.e. $f_\theta(X) = [h_{\text{res}}(X_{\text{ecg}}), h_{\text{lin}}(X_{\text{clin}})]$, where θ represents the parameters of both h_{res} and h_{lin}. Here, h_{res} is a residual neural network [105] with five residual blocks, similar to the one used by Ribeiro et al. [203], whereas h_{lin} is a simple two layer feedforward network. The kernel size of the convolutional layers of h_{res} was adjusted to reflect the higher sample rate of 500 Hz. We chose this residual architecture to process the ECG data as Ribeiro et al. [203] demonstrated its efficacy in related cardiological prediction tasks. Table 3.3 provides a detailed overview of the sizes of all layers and used dropout rates.

Table 3.4: Parameter grid to determine multi-task regularisation parameters. η_{best} refers to the best learning rate from the first selection step.

Parameter	Values
λ_{MPSSRS}	$\{0.0, 0.25, 0.5, 0.75, 1.0\}$
λ_{MPSSSS}	$\{0.0, 0.25, 0.5, 0.75, 1.0\}$
λ_{Stress}	$\{0.0, 0.25, 0.5, 0.75, 1.0\}$
η	$\{2\eta_{\text{best}}, \eta_{\text{best}}, \eta_{\text{best}}/2\}$

Lead and Parameter Selection To evaluate the impact that individual ECG leads, preprocessing, and auxiliary tasks have on predictive performance, we proceeded as follows: First, we used the first development split to determine the most promising leads (in terms of area under the precision-recall curve (AUPRC) on the validation set) by performing a grid search over a) three preprocessing schemes described above, and b) learning rate parameters $\eta \in \{0.01, 0.001, 0.0001\}$ for all twelve leads individually and in combination. Subsequently, we picked the three best-performing leads and their respective preprocessing/learning rate combination to assess the impact of all auxiliary tasks. In order to do so, the performance on the validation set was averaged over *all* splits on a $5 \times 5 \times 5 \times 3$ parameter grid as shown in Table 3.4. Finally, the best-performing model was enriched with clinical data to receive the final model, which we evaluated on the held-out test set.

3.3.2.4 Performance Assessment & Comparison Partners

When assessing a decision support system, coarse-grained metrics such as AUROC or AUPRC may not be sufficient to draw a full picture of its clinical value. While the area under the ROC curve provides the probability of ranking a randomly-selected positive sample (i.e. a patient with EIMI) higher than a negative one (i.e. a patient without EIMI) [81], in practice it is relevant to ensure a high baseline sensitivity and/or specificity. We therefore analyse the performance of our system in terms of false positive/negative rates (FPR/FNR) at sensitivity (fraction of correctly predicted positives) and specificity (fraction of correctly predicted negatives) values between 0.90 and 1.0. The FPR is the proportion of all negatives (patients without EIMI) that are predicted to suffer from EIMI. Equivalently, FNR is the fraction of all positives (patients with EIMI) that are predicted to not suffer from EIMI.

Human Baseline During the exercise stress test, we recorded the treating physician's judgement concerning the presence of myocardial ischaemia after the stress test on a visual analogue scale (see Figure 3.9). This score provides an important human baseline since it is

an indicator as to whether the cardiologist would recommend a follow-up examination with myocardial perfusion SPECT. Suppose a system can reach the predictive performance of a cardiologist in interpreting a stress test. In that case, it will be particularly useful in settings in which specialists are not available. These environments are of tremendous importance because, in general, a stress test does not necessitate the presence of a cardiologist and can be performed by general practitioners. We will refer to this judgement as "Post-Test VAS".

ST-Segment Depression ST-segment depression is a morphological feature that is commonly linked to ischaemia [195, 246]. However, the exact time points in the ECG at which ST-amplitude is measured varies [196]. We compute ST-segment depression as follows. First, we perform a QRS-delineation using the "neurokit2" software package [167] on the complete stress test ECG. Then, we determine the mean isoelectric line for each stress phase of a given "2-6-2" sequence. For this, we take the mean of the last l_{PR} milliseconds preceding the Q-wave over all heartbeats in a specific stress phase. Similarly, we determine the mean ST-amplitude for each "2-6-2" stress phase by using the ECG measurement 60 ms after the J-point. The mean ST-segment depression (difference between mean isoelectric line and ST-amplitude) is determined for each stress phase (ST_{Pre}, $\text{ST}_{\text{Stress}}$, ST_{Rec}). The differences between $\text{ST}_{\text{Stress}}$/$\text{ST}_{\text{Rec}}$ and baseline ST-depression (ST_{Pre}) are then aggregated over all "2-6-2" sequences of a patient by using either their mean, median, minimum, or maximum. Importantly, the physiological response to stress may differ among the three subcohorts (see Section 3.3.1.1). Therefore, the parameter grid shown in Table 3.5a is evaluated separately for all three cohorts (see list of relevant subcohorts and their description on page 68) and all leads.

Static Variable Baseline To evaluate the classification performance of a machine learning model only trained on static clinical features, we use a random forest (RF) [38] classifier. It is a powerful non-parametric algorithm able to handle measurements of different scales such as nominal scales (e.g. previous CAD or sex) or ratio scales (e.g. weight or height) without the need of any preprocessing. It has been successfully used in a variety of healthcare-related classification tasks such as disease risk prediction [129] or the identification of high-need and high-cost patients [189]. A grid search is performed over the parameters shown in Table 3.5b. To make use of the fact that random forests are effective learners of tabular data, we also combine its scores with the predictions from `NIP` to evaluate whether the RF learns from clinical data in a way that complements the neural network. For this, we use both scores (`NIP` and RF) and train a logistic regression on the training data set.

Table 3.5: Parameters grids for ST-segment depression and random forest baselines.

(a) Parameter grid for ST-segment depression baseline.

Parameter	Values
Difference computation	$\{\mathrm{ST}_{\mathrm{Stress}} - \mathrm{ST}_{\mathrm{Pre}}, \mathrm{ST}_{\mathrm{Rec}} - \mathrm{ST}_{\mathrm{Pre}}\}$
Difference aggregation	{mean, median, min, max}
l_{PR}	{20 ms, 40 ms, 100 ms}

(b) Parameter grid for random forest baseline. Full maximum depth means nodes are expanded until all leaves are pure.

Parameter	Values
Num. Trees	{25, 50, 100}
Max. num. features	{2, 5, 8}
Max. depth	{full, 5, 10}
Min. impurity increase	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}\}$

3.3.3 Results

First, we report results of the lead and parameter selection experiment to determine preprocessing, input data, and the parameters of the final model. As we will show in Section 3.3.3.1, the best performing lead for the task is lead `V6` when we preprocess the signal with the most thorough preprocessing scheme. Furthermore, we show that a combination of all auxiliary tasks improves predictive performance over non-regularised training. In Section 3.3.3.2, predictive performance and clinical relevance on the held-out test set is presented and followed by an investigation into trustworthiness and interpretability of the proposed model. Throughout this section, we refer to the "Post-Test VAS" score as the human baseline, as it reflects the cardiologist's assessment as to whether a patient is at risk for EIMI before any imaging is ordered. We use this score to augment `NIP`'s purely algorithmic predictions with expert experience by combining both as features for a logistic regression.

3.3.3.1 Lead & Parameter Selection

We trained $13 \times 3 \times 3 = 117$ neural networks to determine the three best performing leads on the first split of the development data set. The first number accounts for the 12 individual ECG leads plus one configuration that combines all leads. The second number represents three preprocessing schemes and is followed by the number of learning rates that were analysed. For the top three leads, we then performed a grid search as described in Sec-

Table 3.6: Impact of regularisation strength on mean AUPRC (%) over all splits and learning rates. Uncertainty is shown as standard deviation. "None" refers to training without any regularisation, "Best" to the configuration with highest mean AUPRC. Highest AUPRC is reached on lead V6 with $\lambda_{\text{MPSSRS}} = \lambda_{\text{MPSSSS}} = 0.5$, and $\lambda_{\text{Stress}} = 0.75$.

Regularisation		Lead aVR	Lead V1	Lead V6
λ_{MPSSRS}	0.0	54.71 ± 1.73	52.47 ± 1.01	55.94 ± 1.26
	0.25	55.55 ± 0.87	52.99 ± 0.45	56.57 ± 0.60
	0.5	55.59 ± 0.86	52.93 ± 0.46	56.70 ± 0.50
	0.75	55.56 ± 0.86	52.93 ± 0.48	56.81 ± 0.45
	1.0	55.26 ± 1.07	52.85 ± 0.48	56.80 ± 0.47
λ_{MPSSSS}	0.0	53.91 ± 1.36	52.07 ± 0.83	55.86 ± 1.18
	0.25	55.56 ± 0.88	53.03 ± 0.40	56.82 ± 0.53
	0.5	55.82 ± 0.75	53.10 ± 0.39	56.90 ± 0.48
	0.75	55.71 ± 0.77	53.04 ± 0.38	56.68 ± 0.48
	1.0	55.66 ± 0.77	52.93 ± 0.40	56.54 ± 0.54
λ_{Stress}	0.0	54.82 ± 1.05	52.64 ± 0.64	56.10 ± 0.76
	0.25	55.37 ± 1.12	52.77 ± 0.61	56.45 ± 0.73
	0.5	55.36 ± 1.16	52.88 ± 0.63	56.70 ± 0.62
	0.75	55.54 ± 1.17	52.89 ± 0.62	56.74 ± 0.85
	1.0	55.57 ± 1.23	52.98 ± 0.67	56.82 ± 0.75
None		51.21 ± 0.17	50.73 ± 0.58	53.80 ± 0.21
Best				57.23 ± 0.68

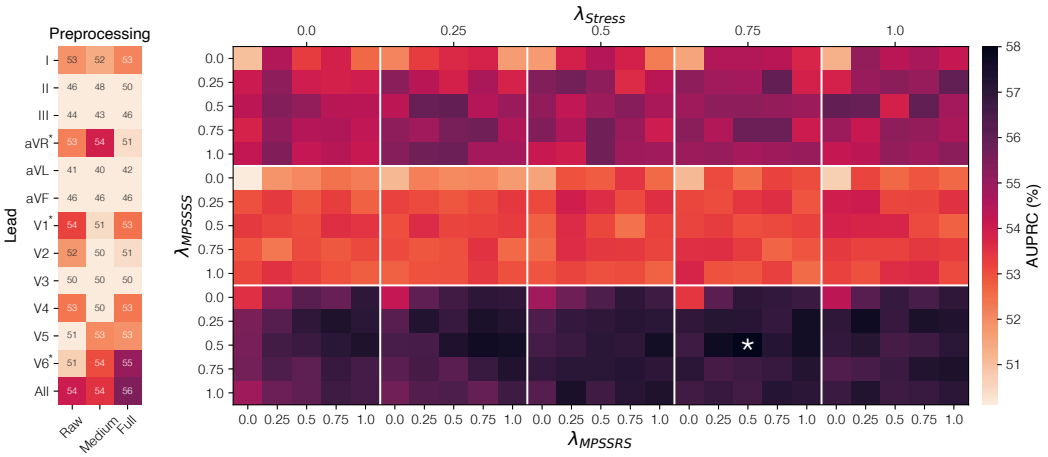


Figure 3.12: Performance heatmaps for lead, preprocessing, and regularisation parameter selection. Prevalence: 34 %. Left: Best AUPRC among three learning rates per lead and preprocessing pipeline. The best three leads are marked with a black asterisk. Right: Results of the grid search to find the best regularisation parameters. Large rows (separated by white lines) represent the three best performing leads (`aVR`, `V1`, `V6`). Large columns represent five settings for λ_{Stress} (upper x-axis), small columns and rows respective regularisation values for λ_{MPSSRS} and λ_{MPSSSS}. The best regularisation combination is marked with a white asterisk.

tion 3.3.2.3 over all splits leading to a total of 5625 trained networks. The results of both grid searches are visualised in Figure 3.12. The best performing leads are `aVR`, `V1`, and `V6` under the medium, raw, and full preprocessing scheme, respectively. Averaged over all ECG leads, the full preprocessing pipeline results in a slight performance increase of 0.81/1.19 percentage points (AUPRC/AUROC) compared to no preprocessing. The moderate preprocessing scheme, however, even leads to a small performance drop of 0.28 in AUPRC and 0.58 in AUROC over all leads. The effect of preprocessing is more pronounced in individual leads. The highest improvements in AUPRC (our selection criterion to determine the model's parameters) can be observed in leads `I` (+3.89) and `V6` (+2.10) for the medium and full preprocessing scheme, respectively. Using all ECG signals as input to the residual network slightly increases predictive performance, but in the spirit of Occam's Razor (and given the magnitude of the increase), we decided to excluded the evaluation of this setting in the following steps.

The impact of different regularisation settings is shown on the right hand side of Figure 3.12. For each configuration, we show the mean AUPRC (over all five splits) of the best performing learning rate and use a white asterisk to highlight the best setting. The most striking difference can be seen in the overall performance of lead `V1` (large middle row) yielding a mean AUPRC of 52.89 %. On average, both other leads perform better with

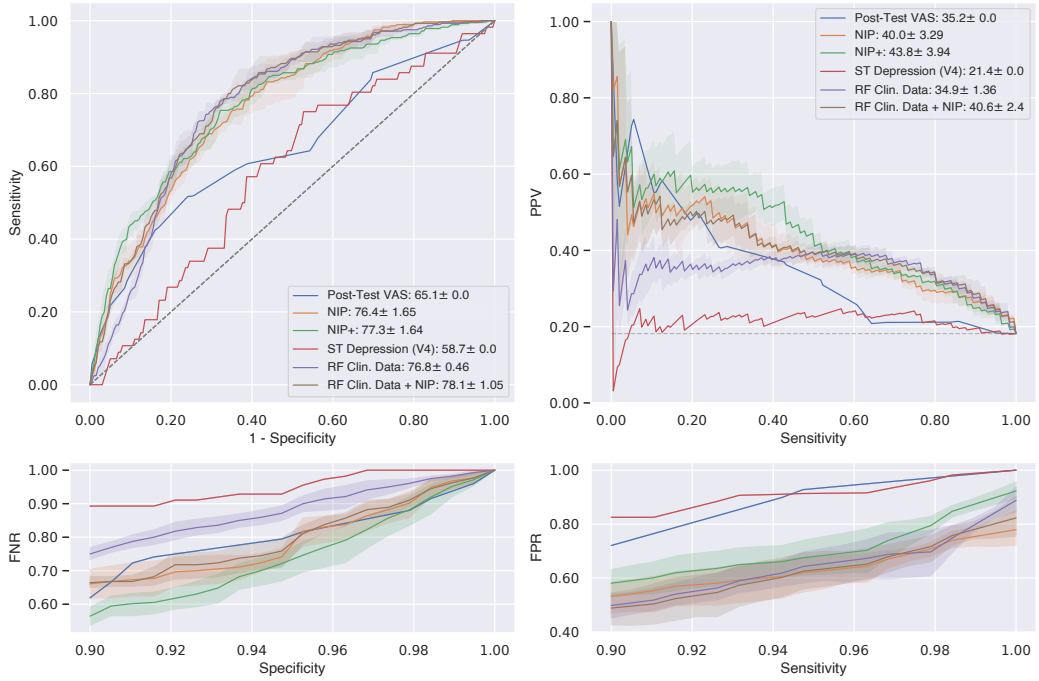


Figure 3.13: Performance overview on a subcohort without previous coronary artery disease. All patients in this subcohort were able to complete the exercise stress test without the need for any pharmacological intervention. The key contains the name of the method and its area under the curve in percentage.

AUPRCs of 55.93 % (aVR) and 56.92 % (V6). Moreover, the importance of the MPS auxiliary tasks are underlined by the performance drop in configurations in which both tasks are "muted" ($\lambda_{\text{MPSSRS}} = \lambda_{\text{MPSSSS}} = 0$). Table 3.6 provides a fine-grained overview of this ablation study. While the differences between individual regularisation strengths are marginal, we observe that each lead reaches its highest performance when being regularised. Lastly, the best performance is obtained on lead V6, setting $\lambda_{\text{MPSSRS}} = \lambda_{\text{MPSSSS}} = 0.5$, $\lambda_{\text{Stress}} = 0.75$, and using a learning rate of 0.0005. This is the setting used for all subsequent experiments.

3.3.3.2 Predictive Performance & Clinical Relevance

After determining the best model, we evaluated its performance on the internal held-out test set containing 874 patients. We performed our analyses on different subcohorts to shed light on the robustness of the model over different input distributions. Figure 3.13 depicts predictive performance of all comparison partners on a subcohort of patients who have no history of CAD and were able to complete the stress test without pharmacological support. The odds of suffering from EIMI are significantly increased ($p = 2.26 \times 10^{-40}$) for patients

Table 3.7: Reduction (negative sign) and increase (positive sign) of mean FPR/FNR at high sensitivity/specificity values with respect to the human baseline. Asterisks indicate the significance level (0.05 and 0.01) at which the difference is statistically significant. Statistical analysis is based on a Kolmogorov-Smirnov [240] one-sample test and is corrected for multiple hypotheses using Bonferroni correction. The sensitivity/specificity values at which any method shows both significant decreases in FPR and FNR are marked in bold.

(a) FPR and FNR reduction for the subcohort that has no history of CAD and was able to perform the complete exercise stress test without requiring pharmacological support.

	NIP		NIP+		RF		RF + NIP	
Sens./Spec.	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR
0.91	-0.21^{*}	−0.05	−0.16	-0.12^{*}	-0.25^{**}	+0.06	-0.26^{*}	-0.06^{*}
0.93	-0.26^{*}	−0.06	−0.20	−0.12	-0.26^{**}	$+0.07^{*}$	-0.28^{*}	−0.04
0.95	-0.31^{**}	0.0	-0.25^{*}	−0.07	-0.28^{**}	$+0.08^{*}$	-0.30^{**}	−0.01
0.97	-0.27^{**}	+0.01	-0.20^{**}	−0.01	−0.27	$+0.08^{*}$	-0.27^{**}	+0.02
0.99	-0.23^{**}	+0.02	-0.09^{*}	+0.01	-0.15^{**}	$+0.03^{**}$	-0.19^{**}	+0.02

(b) FPR and FNR reduction for the complete internal held-out data set.

	NIP		NIP+		RF		RF + NIP	
Sens./Spec.	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR
0.91	-0.13^{**}	+0.03	-0.16^{**}	+0.01	-0.17^{**}	$+0.07^{**}$	-0.17^{**}	+0.01
0.93	-0.14^{**}	−0.01	-0.17^{**}	−0.03	-0.15^{**}	+0.03	-0.16^{**}	−0.02
0.95	-0.15^{**}	0.00	-0.18^{**}	−0.02	-0.15^{**}	+0.03	-0.17^{**}	−0.01
0.97	-0.15^{*}	0.00	-0.13^{*}	−0.03	-0.1^{**}	+0.04	-0.14^{*}	0.0
0.99	-0.13^{**}	+0.01	-0.10^{**}	+0.01	-0.07^{*}	+0.01	-0.10^{**}	0.0

with previous CAD (odds ratio (OR): 2.64, 95 % confidence interval (CI): 2.28-3.05) over the whole cohort (p-values were computed using Fisher's exact test [84]). Moreover, as we will see in Section 3.3.3.3, existence of previous CAD influences our method the most in predicting a high EIMI score. Similarly, absence of known CAD tends to reduce the predicted EIMI score. By reducing the evaluation cohort to patients without a history of CAD, we prevent all algorithms and the cardiologist from relying on this correlated variable to predict the presence of EIMI.

In Figure 3.13, we show receiver operating characteristic (ROC) and precision-recall (PR) curves in the upper left and right plots, respectively. Each opaque line represents the mean performance of the respective methods when trained on 5 different training splits. Envelopes show standard deviations. In terms of mean predictive performance, we observe that our

proposed method (NIP) and its extension (NIP+), as well as the random forests (RF and RF + NIP) outperform the ST-depression algorithm *and* the human baseline. Furthermore, the addition of the cardiologist's judgement to the neural network approach slightly increases the AUROC from 76.4 % to 77.3 %. This effect is even more pronounced in mean AUPRC with an increase from 40.0 % to 43.8 %. We further observe that the random forest's AUPRC trained on static clinical variables is on par with the expert opinion (at least on average). While showing promising *overall* predictive performance, the weakness of this simple baseline is illustrated in the left lower plot where we investigate predictive performance in terms of FNR at high specificity. This setting is of higher clinical relevance than summary metrics such as AUROC and AUPRC as it is necessary to maintain a high detection rate of both classes. The lower plots of Figure 3.13 are augmented and quantified by Table 3.7a, showing

Table 3.8: Performance analysis on three relevant subcohorts: Patients that completed the stress test on the bicycle, patients on a pharmacological protocol, and patients who needed pharmacological support during the exercise to reach their target heart rate. The first column contains a short description of the subcohort, its size, and the prevalence of EIMI.

Subcohort	Method	AUPRC (%)	AUROC (%)
Full Exercise Test n=481, prev.: 24.5 %	Post-Test VAS	43.86 ± 0.00	67.4 ± 0.0
	NIP	47.16 ± 1.51	74.84 ± 0.99
	NIP+	**50.82 ± 1.80**	**76.92 ± 1.03**
	ST Depression (V4)	28.86 ± 0.00	59.38 ± 0.00
	RF Clin. Data	41.79 ± 1.17	73.43 ± 0.53
	RF Clin. Data + NIP	47.71 ± 1.50	75.44 ± 0.74
Full Pharma Stress n=100, prev.: 33.0 %	Post-Test VAS	37.31 ± 0.00	57.15 ± 0.00
	NIP	45.55 ± 2.46	**69.80 ± 2.77**
	NIP+	**46.18 ± 2.72**	68.84 ± 2.62
	ST Depression (II)	36.04 ± 0.00	53.64 ± 0.00
	RF Clin. Data	42.97 ± 3.04	66.03 ± 1.40
	RF Clin. Data + NIP	45.32 ± 2.50	69.44 ± 1.65
Combined Stress n=221, prev.: 34.4 %	Post-Test VAS	45.42 ± 0.00	57.26 ± 0.00
	NIP	45.76 ± 0.82	63.38 ± 1.38
	NIP+	46.07 ± 1.21	**64.24 ± 1.41**
	ST Depression (V4)	43.60 ± 0.00	59.71 ± 0.00
	RF Clin. Data	40.93 ± 2.10	59.61 ± 0.71
	RF Clin. Data + NIP	**46.14 ± 0.52**	63.48 ± 0.91

the differences (Δ) in mean FPR/FNR with respect to the human baseline (Post-Test VAS score). Here, the random forest (purple) shows a statistically significant *increase* of FNR at all specificity values over the human baseline (blue). This indicates that there are many patients suffering from EIMI that the cardiologist detects, but the algorithm does not, making it a computational approach that does not add any value in the identification of patients at risk (in contrast, `NIP` leads to an FNR *reduction* at 2/5, and `NIP+` at 4/5 specificity thresholds). The most significant benefit of our proposed methods can be observed in the reduction of false positives (lower right plot in Figure 3.13). At a sensitivity of 0.95, `NIP` decreases the FPR of the human baseline from 0.94 to 0.63. `NIP` can therefore be used as a risk stratification tool and help making a decision as to whether a patient should receive a myocardial perfusion SPECT or not. This way, `NIP` can identify up to 31 % of the patients without CAD for which a cardiologist would recommend a myocardial perfusion scan, reducing costs and the patient's exposure to radioactive tracers. A similar, yet not as significant, trend can be observed when we evaluate the performance over the full cohort. Table 3.7b shows results of this analysis. At a sensitivity/specificity of 0.95, `NIP+` continues to reduce the number of false positives significantly, while showing a slight, non-significant drop in FNR.

3.3.3.3 Trust and Interpretability

Schlesinger and Stultz [225] stressed the importance of trustworthiness and interpretability of risk stratification models in cardiology. According to the authors, having trust in a model means to understand whether it performs well on a physician's specific patient. It is therefore crucial to identify cohorts of the population for which the model performs particularly well and especially poorly. Furthermore, for the cardiologist who interacts with the risk model, it is critical to understand what precisely the model "has learnt" and whether its internal representation is consistent with the physician's knowledge about the phenotype. To address the issue of trust, we evaluate the model's performance on a variety of subcohorts that are important in the context of exercise stress testing. Regarding interpretability, we perform an analysis of SHAP values [163] on population level and a case study on a selected patient to illustrate what our model "has learnt".

Table 3.9: Detailed performance analysis on patients who underwent full exercise test.

Subcohort	Method	AUPRC (%)	AUROC (%)
Female Full Exercise Test No Prev. CAD n=131, prev.: 6.1 %	Post-Test VAS	**26.31 ± 0.00**	49.60 ± 0.00
	NIP	15.37 ± 3.10	71.88 ± 1.02
	NIP+	18.95 ± 5.01	64.74 ± 2.96
	ST Depression (V4)	7.61 ± 0.00	52.26 ± 0.00
	RF Clin. Data	16.23 ± 4.34	71.85 ± 4.09
	RF Clin. Data + NIP	16.46 ± 4.44	**72.34 ± 1.84**
Male Full Exercise Test No Prev. CAD n=177, prev.: 27.1 %	Post-Test VAS	42.55 ± 0.00	66.55 ± 0.00
	NIP	45.54 ± 3.26	70.52 ± 1.74
	NIP+	**49.47 ± 3.72**	**73.08 ± 1.72**
	ST Depression (V4)	29.65 ± 0.00	56.87 ± 0.00
	RF Clin. Data	36.39 ± 1.57	66.99 ± 1.72
	RF Clin. Data + NIP	44.76 ± 1.56	71.20 ± 1.13
Female Full Exercise Test Prev. CAD n=32, prev.: 18.8 %	Post-Test VAS	**59.18 ± 0.00**	**75.64 ± 0.00**
	NIP	23.34 ± 4.20	57.87 ± 6.33
	NIP+	28.06 ± 1.67	63.50 ± 3.89
	ST Depression (V4)	17.84 ± 0.00	43.55 ± 0.00
	RF Clin. Data	46.27 ± 7.19	72.96 ± 4.82
	RF Clin. Data + NIP	27.61 ± 7.76	60.65 ± 6.44
Male Full Exercise Test Prev. CAD n=141, prev.: 39.7 %	Post-Test VAS	58.07 ± 0.00	68.80 ± 0.00
	NIP	56.34 ± 0.67	65.88 ± 0.76
	NIP+	**60.27 ± 1.15**	**69.79 ± 0.70**
	ST Depression (V4)	47.19 ± 0.00	61.30 ± 0.00
	RF Clin. Data	46.19 ± 1.77	56.14 ± 1.64
	RF Clin. Data + NIP	56.63 ± 0.72	66.13 ± 0.65

Table 3.10: Detailed performance analysis on patients who were not able to exercise and underwent complete pharmacologically-induced stress.

Subcohort	Method	AUPRC (%)	AUROC (%)
Female Full Pharma Stress No Prev. CAD n=24, prev.: 8.3 %	Post-Test VAS	6.74 ± 0.00	40.91 ± 0.00
	NIP	13.88 ± 2.82	71.71 ± 4.22
	NIP+	11.87 ± 5.90	59.95 ± 6.20
	ST Depression (II)	16.51 ± 0.00	**79.40 ± 0.00**
	RF Clin. Data	**18.21 ± 17.99**	58.14 ± 10.52
	RF Clin. Data + NIP	13.17 ± 4.05	68.09 ± 7.87
Male Full Pharma Stress No Prev. CAD n=19, prev.: 42.1 %	Post-Test VAS	44.17 ± 0.00	53.95 ± 0.00
	NIP	**62.58 ± 8.70**	64.02 ± 6.80
	NIP+	62.03 ± 7.87	**67.86 ± 7.98**
	ST Depression (II)	34.56 ± 0.00	39.82 ± 0.00
	RF Clin. Data	43.09 ± 5.48	54.75 ± 8.76
	RF Clin. Data + NIP	61.46 ± 7.43	63.57 ± 6.14
Female Full Pharma Stress Prev. CAD n=13, prev.: 30.8 %	Post-Test VAS	29.19 ± 0.00	58.33 ± 0.00
	NIP	**82.82 ± 6.80**	**83.72 ± 8.13**
	NIP+	78.68 ± 5.76	77.09 ± 5.39
	ST Depression (II)	30.44 ± 0.00	52.76 ± 0.00
	RF Clin. Data	39.65 ± 9.21	59.40 ± 7.16
	RF Clin. Data + NIP	80.83 ± 7.31	83.17 ± 8.05
Male Full Pharma Stress Prev. CAD n=44, prev.: 43.2 %	Post-Test VAS	48.03 ± 0.00	**59.91 ± 0.00**
	NIP	42.33 ± 1.49	52.60 ± 2.79
	NIP+	42.72 ± 2.66	53.32 ± 2.17
	ST Depression (II)	**57.26 ± 0.00**	56.35 ± 0.00
	RF Clin. Data	43.97 ± 5.32	51.44 ± 4.03
	RF Clin. Data + NIP	42.12 ± 1.74	51.71 ± 2.63

Table 3.11: Detailed performance analysis on patients who started the stress test on the bicycle but needed pharmacological support to reach their target heart rate.

Subcohort	Method	AUPRC (%)	AUROC (%)
Female Combined Stress No Prev. CAD n=50, prev.: 20.0 %	Post-Test VAS	20.07 ± 0.00	**58.50 ± 0.00**
	NIP	18.92 ± 2.06	50.20 ± 3.69
	NIP+	19.33 ± 1.79	52.36 ± 2.99
	ST Depression (V4)	22.23 ± 0.00	53.02 ± 0.00
	RF Clin. Data	**23.42 ± 5.31**	47.97 ± 2.34
	RF Clin. Data + NIP	18.97 ± 2.04	48.64 ± 3.32
Male Combined Stress No Prev. CAD n=45, prev.: 35.6 %	Post-Test VAS	**62.72 ± 0.00**	60.16 ± 0.00
	NIP	47.22 ± 3.63	63.61 ± 3.88
	NIP+	50.49 ± 1.82	64.53 ± 3.54
	ST Depression (V4)	52.80 ± 0.00	58.83 ± 0.00
	RF Clin. Data	39.99 ± 2.89	58.41 ± 2.52
	RF Clin. Data + NIP	47.80 ± 2.45	**64.69 ± 2.59**
Female Combined Stress Prev. CAD n=22, prev.: 22.7 %	Post-Test VAS	**35.86 ± 0.00**	48.86 ± 0.00
	NIP	27.45 ± 5.22	**58.38 ± 2.42**
	NIP+	27.30 ± 5.13	56.45 ± 4.21
	ST Depression (V4)	22.07 ± 0.00	54.12 ± 0.00
	RF Clin. Data	18.95 ± 4.24	40.73 ± 7.68
	RF Clin. Data + NIP	23.31 ± 1.53	54.16 ± 6.36
Male Combined Stress Prev. CAD n=104, prev.: 43.3 %	Post-Test VAS	49.54 ± 0.00	56.44 ± 0.00
	NIP	52.83 ± 2.15	61.07 ± 2.41
	NIP+	52.46 ± 1.91	**62.71 ± 2.06**
	ST Depression (V4)	**57.08 ± 0.00**	62.54 ± 0.00
	RF Clin. Data	43.59 ± 3.74	48.53 ± 3.16
	RF Clin. Data + NIP	53.09 ± 1.71	60.91 ± 2.53

Subcohort Analysis The three main subcohorts in a stress test are 1. patients who underwent pure exercise stress test, 2. patients who started exercising but needed pharmacological support to reach their target heart rate, and 3. patients who were not able to exercise altogether and for whom a full pharmacological protocol was executed. Table 3.8 shows that in all three subcohorts, methods that include a "NIP component" always outperform the other methods in terms of AUROC and AUPRC. We observe the largest increase over the human

baseline (Post-Test VAS) in the cohort tested using the full pharmacological protocol. The highest performance in terms of AUPRC was achieved in the full exercise cohort in which the gain in AUPRC over the prevalence was also highest (a random classifier will achieve an AUPRC that is equal to the prevalence). Schlesinger and Stultz [225] also pointed out the importance of understanding for which patient population a risk model may underperform. We therefore analysed the subcohorts from above in more detail as shown in Tables 3.9, 3.10, and 3.11. However, as the first cohort (i.e. patients that underwent pure exercise testing) represents the largest number of patients, this subgroup will be the focus of the following discussion. Among the subcohort of patients that underwent full exercise stress testing (Table 3.9), males are overrepresented, making up almost 90 % of the cases. As this was also reflected in the training set, we observe *higher* performance gains in males than in female patients. However, there is still a notable performance increase over the human baseline in terms of AUROC in female patients who have no history of CAD. To further investigate clinically relevant metrics in this subcohort, we performed the same FPR/FNR analysis as in Table 3.7, where we showed the reduction/increase in mean FPR/FNR compared to the human baseline. The results are shown in Tables 3.12, 3.13, and 3.14. A method/threshold pair is considered relevant if the method shows a decrease in either FNR or FPR and no increase in the other metric. If this is the case, both values are highlighted in bold. If all entries of a stratum are relevant, the subcohort description is shown in bold. To compute the delta at the shown sensitivity and specificity levels, we performed a linear interpolation on all methods' FPR/FNR values. From this follows that approaches for which no decision threshold leads to a sensitivity/specificity value higher than 0.91, the linear imputation of FPR/FNR may result in an inaccurate delta. If this was the case for the Post-Test VAS score, entries are shown in grey. Focusing on black bold entries only, the largest decrease in FNR combined with a significant reduction in FPR was achieved by `NIP+` at a sensitivity/specificity threshold of 0.91 for male patients with no history of CAD who were able to complete the exercise test. Moreover, we observe that `NIP+` shows the highest number of relevant pairs (23) over all subcohorts, followed by `NIP` (18), underlining the relevance of a collaborative approach where network scores are combined with a physician's judgement.

Table 3.12: Performance analysis on patients who underwent full exercise stress testing. A method is considered relevant and marked in bold if for a given sensitivity/specificity one metric (FPR or FNR) is decreased while to other is at least not increased. Grey values indicate that the results may be inaccurate due to interpolation from sensitivities/specificities smaller than 0.91 for Post-Test VAS.

		NIP		NIP+		RF		RF + NIP	
Subcohort	Sens./Spec.	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR
Female Full Exercise Test No Prev. CAD	0.91	−0.42**	+0.12**	**−0.20****	**−0.03**	−0.47**	+0.12	−0.41**	+0.10
	0.93	−0.41**	+0.14**	−0.18**	+0.01	−0.42**	+0.18*	−0.39**	+0.11**
	0.95	−0.41**	+0.23**	−0.17**	+0.07	−0.36**	+0.22	−0.37**	+0.21**
	0.97	−0.40**	+0.12**	−0.15*	+0.07	−0.30**	+0.11	−0.35**	+0.11
	0.99	−0.40*	+0.02	−0.13	+0.02	−0.24	+0.02	−0.33	+0.02
Male Full Exercise Test No Prev. CAD	0.91	**−0.10**	**−0.09**	**−0.16***	**−0.11**	−0.13	+0.11*	**−0.15***	**−0.04**
	0.93	**−0.12**	**0.0**	**−0.15**	**−0.07**	−0.14	+0.11	−0.15	+0.02
	0.95	**−0.14**	**0.0**	**−0.14***	**−0.05**	−0.13	+0.09*	−0.17	+0.02
	0.97	−0.12	+0.04	**−0.14**	**0.00**	−0.10	+0.08**	−0.17	+0.03
	0.99	**−0.16**	**−0.01**	**−0.19**	**−0.01**	**−0.13***	**0.00**	**−0.19**	**−0.01**
Female Full Exercise Test Prev. CAD	0.91	+0.31	+0.38**	+0.21**	+0.33**	0.0	+0.12	+0.24	+0.33*
	0.93	+0.22	+0.43**	+0.15*	+0.43**	−0.04	+0.19**	+0.15	+0.39**
	0.95	+0.12	+0.48**	+0.10	+0.48**	−0.08	+0.31**	+0.07	+0.44**
	0.97	+0.03	+0.31**	+0.04	+0.31**	−0.13	+0.23**	−0.01	+0.29**
	0.99	−0.06	+0.06**	−0.01	+0.06**	−0.17	+0.05**	−0.09	+0.06**
Male Full Exercise Test Prev. CAD	0.91	+0.03	+0.13**	−0.10	+0.11**	+0.11	+0.17**	+0.04	+0.13**
	0.93	+0.03	−0.04	**−0.11**	**−0.06**	+0.08	+0.03	+0.07	−0.04
	0.95	+0.03	−0.07**	**−0.11**	**−0.09****	+0.08	+0.01	+0.05	−0.07**
	0.97	+0.01	−0.07	**−0.07**	**−0.10****	+0.07	+0.01	+0.03	−0.06
	0.99	+0.01	−0.01	**0.00**	**−0.02**	+0.05	0.00	+0.03	−0.01

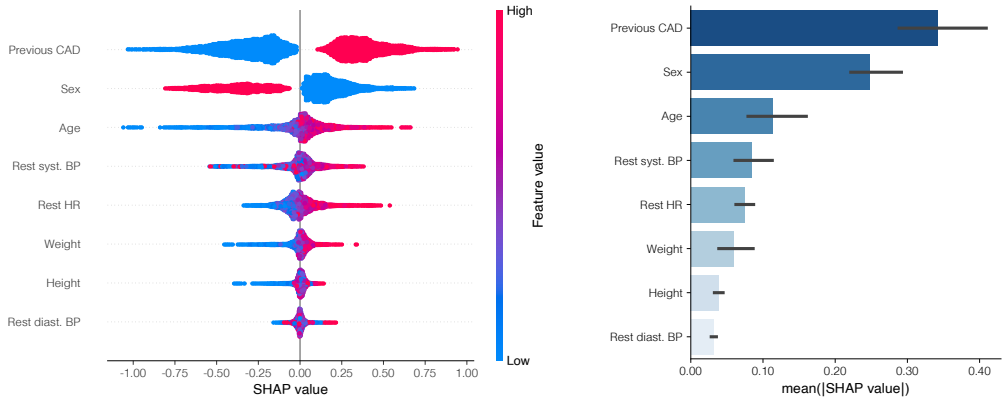
Table 3.13: Performance analysis on patients who underwent full pharmacological stress testing. A method is considered relevant and marked in bold if for a given sensitivity/specificity one metric (FPR or FNR) is decreased while to other is at least not increased. The first column is bold if this is the case for all methods of that subcohort. Grey values indicate that the results may be inaccurate due to interpolation from sensitivities/specificities smaller than 0.91 for Post-Test VAS.

		NIP		NIP+		RF		RF + NIP	
Subcohort	Sens./Spec.	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR
Female Full Pharma Stress No Prev. CAD	0.91	−0.22**	0.0	−0.19*	−0.09	−0.13	−0.10	−0.22**	0.0
	0.93	−0.30**	0.0	−0.22**	−0.05	−0.18	−0.10	−0.27*	0.0
	0.95	−0.37**	0.0	−0.24**	−0.01	−0.23	−0.10	−0.33**	0.0
	0.97	−0.45**	0.0	−0.27*	0.0	−0.28	−0.06	−0.39**	0.0
	0.99	−0.53**	0.0	−0.29	0.0	−0.33	−0.01	−0.45**	0.0
Male Full Pharma Stress No Prev. CAD	0.91	−0.07	−0.28	−0.16	−0.17	−0.10	−0.02	−0.06	−0.24
	0.93	−0.14	−0.22	−0.19	−0.13	−0.14	−0.02	−0.13	−0.18
	0.95	−0.21	−0.15	−0.22	−0.09	−0.17	−0.01	−0.19	−0.12
	0.97	−0.28	−0.08	−0.24	−0.05*	−0.21	−0.01	−0.25	−0.07
	0.99	−0.35*	−0.02	−0.27	−0.01	−0.25*	0.00	−0.31**	−0.01
Female Full Pharma Stress Prev. CAD	0.91	−0.54**	−0.60**	−0.47	−0.55**	−0.19	−0.08	−0.49*	−0.55**
	0.93	−0.46	−0.46**	−0.35	−0.42**	−0.18	−0.07	−0.43	−0.42**
	0.95	−0.38	−0.31**	−0.24	−0.29**	−0.16	−0.04	−0.37	−0.29**
	0.97	−0.30	−0.18**	−0.12	−0.16**	−0.15	−0.03	−0.32	−0.16**
	0.99	−0.22	−0.03**	−0.01	−0.03**	−0.14	−0.01	−0.26	−0.03**
Male Full Pharma Stress Prev. CAD	0.91	+0.01	+0.15**	+0.07	+0.13	+0.18*	+0.11	+0.06	+0.15**
	0.93	+0.02	+0.05	+0.07	+0.06	+0.17**	+0.04	+0.08	+0.05
	0.95	+0.02	+0.01	+0.06	+0.02	+0.17**	0.0	+0.10	+0.01
	0.97	+0.01	−0.01	+0.05	0.0	+0.16**	−0.01	+0.07	−0.01
	0.99	**−0.01**	**0.00**	+0.04	0.00	+0.16**	0.00	+0.05	0.00

Table 3.14: Performance analysis on patients who underwent combined stress testing. A method is considered relevant and marked in bold if for a given sensitivity/specificity one metric (FPR or FNR) is decreased while to other is at least not increased. Grey values indicate that the results may be inaccurate due to interpolation from sensitivities/specificities smaller than 0.91 for Post-Test VAS.

		NIP		NIP+		RF		RF + NIP	
Subcohort	Sens./Spec.	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR	Δ FPR	Δ FNR
Female Combined Stress No Prev. CAD	0.91	+0.18*	−0.02	+0.09	−0.02	+0.29**	−0.06	+0.23**	−0.02
	0.93	+0.16**	−0.01	+0.06	0.0	+0.25**	−0.04	+0.21**	0.0
	0.95	+0.15*	0.0	+0.03	0.0	+0.21*	−0.04	+0.18**	0.0
	0.97	+0.13*	0.0	**−0.01**	**0.0**	+0.17	−0.04	+0.16**	0.0
	0.99	+0.11	0.0	**−0.04**	**0.0**	+0.13	−0.01	+0.14*	0.0
Male Combined Stress No Prev. CAD	0.91	−0.18	+0.31**	−0.19	+0.27**	−0.16**	+0.38**	−0.19	+0.32**
	0.93	−0.13	+0.31**	−0.10	+0.28**	−0.15**	+0.38**	−0.15	+0.32**
	0.95	−0.09	+0.32**	−0.05	+0.30**	−0.11**	+0.37**	−0.12	+0.32**
	0.97	−0.07	+0.23**	−0.04	+0.22**	−0.07	+0.26**	−0.10	+0.23**
	0.99	−0.05	+0.05**	−0.03	+0.04**	−0.04	+0.05**	−0.08	+0.05**
Female Combined Stress Prev. CAD	0.91	−0.49**	+0.13	−0.38*	+0.13	−0.24*	+0.17*	−0.44**	+0.15
	0.93	−0.45**	+0.15	−0.34**	+0.15	−0.23	+0.19**	−0.39*	+0.18**
	0.95	−0.41**	+0.12	−0.30**	+0.12	−0.21	+0.15**	−0.34*	+0.15**
	0.97	−0.37**	+0.07	−0.26*	+0.07	−0.19	+0.08**	−0.29*	+0.08**
	0.99	−0.33	+0.01	−0.22	+0.01	−0.17*	+0.02**	−0.25	+0.02**
Male Combined Stress Prev. CAD	0.91	−0.12	+0.04	−0.13	+0.09**	−0.01	+0.10**	−0.09	+0.05*
	0.93	−0.12	+0.01	−0.13	+0.05	−0.02	+0.05*	−0.09	+0.02
	0.95	**−0.11**	**0.00**	−0.10	+0.03	−0.02	+0.01	**−0.08**	**−0.01**
	0.97	**−0.09**	**0.0**	−0.08	+0.01	**−0.02**	**0.0**	**−0.05**	**−0.01**
	0.99	**−0.08**	**0.00**	**−0.06****	**0.00**	**−0.03***	**0.00**	**−0.05**	**0.00**

Interpretability According to Biran and Cotton [22], a machine learning system is interpretable if a human can understand its operations. This understanding may be developed through introspection or through a produced explanation. Providing useful explanations for predictions is crucial for user acceptance of a decision support system, particularly in a healthcare context [258]. The reasons why interpretable machine learning models are desirable are manifold and Lipton provides a useful taxonomy [156]. In addition to building trust in the model, desiderata of machine learning interpretability are the inference of causal relationships in the real world, transferability of a model to unfamiliar situations, the provision of useful information to a human decision maker, and the assessment of fairness and ethics in automated decision-making. Lipton not only answers the question as to *why* a machine learning model should be interpretable but also provides two model properties that comprise or enable interpretations: Transparency and post-hoc interpretability. In this section, we focus on post-hoc interpretability and provide the physician with visualisations that shed



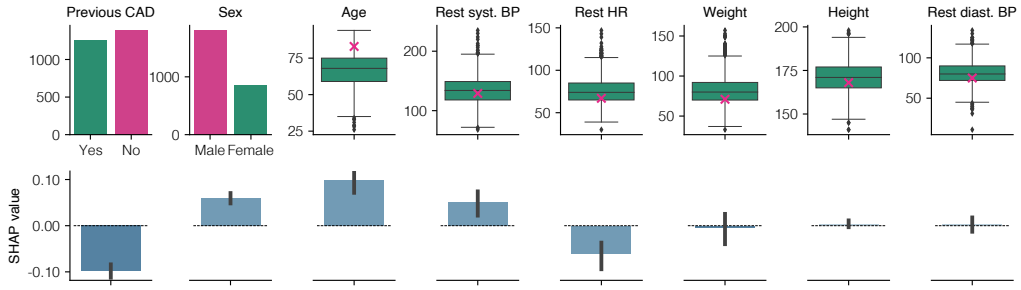
(a) Distribution of SHAP values on the clinical variables used for prediction. Blue/red values indicate low/high features values. Presence/absence of previous CAD are encoded with 1/0. Female/male sex is encoded with 2/1.

(b) Absolute SHAP values per clinical variable. Grey bars show the 95 % confidence interval.

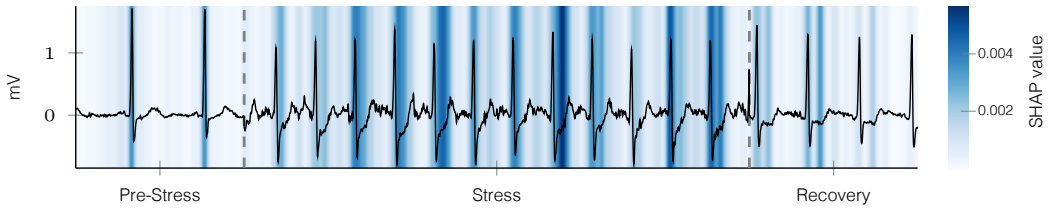
Figure 3.14: SHAP values for all clinical variables computed on the held-out test set.

light on the features that were the driving factors leading to a predicted outcome. This way, the cardiologist can assess whether the way the model "sees" a patient is consistent with their knowledge about the pathology.

To do so, we utilise SHapley Additive exPlanations (SHAP) [163], a game-theoretic approach to explain the outputs of machine learning models. SHAP values provide a score that measures the contribution of each feature to the prediction. A positive SHAP value indicates that a given feature contributes to the prediction of the positive class (presence of EIMI). Conversely, a feature with negative SHAP value influences the model towards predicting the negative class (absence of EIMI). Figure 3.14a depicts the SHAP values of each clinical variable from all patients of the held-out test set. Each dot represents the mean SHAP value over all five splits. History of CAD and sex show a distinct separation, where the binary values both variables can take *always* lead to either increased or decrease SHAP values. The individual contributions of both features range between comparatively high values of $-1.05/-0.80$ (CAD history) and 0.66/0.68 (sex), respectively. More precisely, presence of CAD or being male consistently contributes to a higher predicted risk score for suffering from EIMI. A similar, yet less distinctive pattern can be observed for the age variable. In some cases, the age of younger patients may even reduce the predicted risk for EIMI more than a history of CAD. In addition to the "direction" a feature contributes to the predicted score, Figure 3.14b shows a feature's *absolute* impact. This analysis underlines the importance of CAD history and sex as predictive features. Using Welch's t-test for independent samples,



(a) Upper row: Distribution of clinical variables in the training set (green) and respective values taken by the chosen case (pink). Lower row: SHAP values for each variable of the selected patient.



(b) "2-6-2" sequence for the selected patient with SHAP values for each individual measurement (blue). Higher SHAP values accumulate around R-peaks and ST-segments of the stress-phase. In the pre-stress phase, where almost no ST-depression is visible, SHAP values around the ST-segment are close to zero.

Figure 3.15: SHAP value case study of a 83 year old patient with no history of CAD and a predicted risk score of 0.77. The three clinical variables contributing the most to an increased score are sex, age, and systolic blood pressure at rest.

both variables exhibit significantly higher SHAP values ($p = 0.003$ for gender, 0.0008 for CAD history) compared to the age variable.

In addition to performing a population-wide feature relevance analysis, SHAP values also allow for sample-specific analyses. In Figure 3.15, we show a case study of a 83 year old male patient with no previous CAD. The first row of Figure 3.15a depicts the distributions of all clinical features from the training population. In pink, we show where the patient lies with respect to the training distribution (i.e. the distribution the network "knows"). The second row shows the distribution of SHAP values over five iterations. Moreover, we show the influence of individual measurements of the input ECG in Figure 3.15b. We only show positive SHAP values to get a better understanding which ECG patterns the network associates with ischaemia. The mean risk-score `NIP` provides for this ischaemic patient is 0.77 (1.0 is the maximum score that can be reached; 0.0 the minimum). Notwithstanding the "wrong" direction, among the clinical variables, the absence of a previous CAD contributes the most to the model's prediction (mean SHAP value of −0.10). A signal in the same direction (mean SHAP value of −0.07) is provided by the comparatively low resting heart rate of 67 BPM. While weight, height and diastolic blood pressure influence the model only marginally (and

in both directions), the fact that the patient is male contributes the most towards a higher risk score (positive SHAP value). Similarly, the patient's age, which lies above the upper quartile of the training distribution, pushes the model towards a higher score. Lastly, the *slightly* elevated systolic blood pressure (129 mmHg) also contributes to the prediction of the positive class. The largest contribution that increases the model's output comes from the ECG. The mean SHAP value for the whole signal is 2.31. Figure 3.15b highlights that some regions of the ECG contribute more to a higher EIMI score than others. The highest SHAP values can be observed in the part of the input signal that comes from the stress phase of the examination. Measurements around the R-peak and more strikingly around the ST-segment in the stress and partially in the recovery phase result in higher SHAP values than other segments of the ECG. The latter observation is a data-driven and a priori domain-agnostic confirmation of the relevance of ST-segment depression in the diagnosis of ischaemia [195, 196, 246]. This is underlined by the fact that in the pre-stress phase, where almost no ST-depression is visible, SHAP values around the ST-segment are close to zero.

3.3.4 Conclusion

In this section we introduced a deep-learning based system for the prediction of exercise-induced myocardial ischaemia. First, we showed that a certain degree of signal preprocessing was necessary to reach high predictive performance, even when using deep learning. We also demonstrated that for the task at hand, the signal of a single lead (`V6`) could be almost as informative as learning from all 12 leads simultaneously. Second, we evaluated the value of multi-task learning through an ablation study that assessed three types of inductive biases related to ischaemia. The best combination yielded a performance increase of almost 5 percentage points in AUPRC. Finally, we showed that, in combination with the expert judgement of a cardiologist, our method was able to reduce false positive *and* false negative rates significantly while maintaining high sensitivity and specificity in a subcohort of patients with no previous CAD. To increase transparency and interpretability of our model, we performed a SHAP value analysis, which allows the treating cardiologist to understand the input and output of the deep learning system better. This way, we exemplified in a case study, that abnormalities of the ST-segment contribute to an increased predicted outcome, an observation that aligns with our current knowledge about the physiological manifestation of ischaemia.

We believe that this work shows how collaborative machine learning approaches can yield significant performance improvements at clinical relevance in the classification of complex cardiac pathologies. While the internal assessment is encouraging, to demonstrate the generalisation capabilities of our system, a validation on an externally acquired data set is necessary. It will be particularly interesting to see how our model is transferable to a stress test

setting that uses treadmill instead of bicycle exercise. For future work, we therefore envision to evaluate `NIP` on the telemetric and Holter ECG warehouse (THEW) [57] data set that provides data from a similar study population. In addition to an external validation, it is crucial to establish the clinical added value of our system in a prospective study. This would entail not only the investigation of the realisation of "promised" reduction in false positives and negatives but also the extent to which the physician's judgement is influenced by a) being provided with a risk score altogether, and b) being provided with the SHAP values that make the algorithm more transparent. Follow-up studies like this are crucial to sustainably reshape the field of cardiology and increase the acceptance of "black-box" machine learning as decision-support systems in healthcare in general.

Part II

Object-Valued Time Series

4 Time-Varying Graphs

In which we analyse time series of graphs as a representation of artificial neural networks using persistent homology.

After focusing on real-valued time series of sensor measurements in the preceding chapters, this last chapter revolves around time series of structured objects. As outlined in the introduction, this view generalises the "classical" perspective on time series and aims to provide a holistic approach to time series modelling. By considering a time-evolving object's properties as a unit and treating them accordingly, we can develop expressive and domain-specific time series analysis algorithms. In this regard, a framework for the principled analysis of object-valued time series is developed at the end of Section 5. Graphs or networks represent a structured data type of particular interest in biomedicine and the life sciences. They appear as protein-protein interaction networks [296], metabolic pathways [64, 171], representations of molecules [247], biomedical knowledge graphs [116], or as representations of neural connectivity in the brain [209]. In the following sections, we will develop a method to investigate the learning behaviour of *artificial* neural networks. Our method is rooted in persistent homology (PH), a technique from the field of topological data analysis that provides tools to describe the structure of manifolds. The relation to object-valued time series is made by viewing the network as a stratified graph whose edges change during training. We show that by measuring a network's structural complexity, we are able to make statements about its predictive performance and generalisation capabilities. The following publication is the foundation of this chapter:

- B. Rieck†, M. Togninalli†, **C. Bock**†, M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt. "Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology". In: *International Conference on Learning Representations (ICLR)*. 2019. DOI: 10.3929/ethz-b-000327207

The next sections are structured in the following way: We first provide a brief and self-contained introduction into the problem setting and persistent homology. Subsequently, we

present a novel complexity measure for artificial neural networks and detail its properties before illustrating its advantages over comparable graph-theoretical measures. In Section 4.3.3, we illustrate a correlation between deep learning best practices and our measure, and evaluate an early stopping criterion that uses the network's structural complexity alone. We end this chapter by highlighting the importance of the field of topological machine learning that emerged over the last years.

4.1 Introduction

In the last chapter, we focused on time series in the "classical" sense: a low-dimensional sequence of observations (e.g. measurements of vital signs). While many objects such as arrowheads or leaves can be treated as time series [288], we continue to analyse objects that change over time. More specifically, this chapter revolves around how the structure of deep artificial neural networks changes during training. To investigate this behaviour, we interpret individual layers of a neural network as stratified graphs whose edge weights are changing as part of the fitting process. Using methods from topological data analysis (TDA), we measure changes of a network's structural complexity while being fitted to shed light on the relationship between generalisability and network structure. This is of specific interest to the machine learning community as the practical and unprecedented successes of deep learning in fields such as biomedicine [50, 125, 199, 200] or language translation [13, 40, 253] continue to outpace our theoretical understanding. Particularly, *formal measures* that shed light on the generalisation capabilities of neural networks are yet to be identified [294].

Hitherto, the focus of approaches for improving our practical and theoretical understanding was on interrogating networks using input data. Such methods include

i) the analysis of sensitivity and relevance of features [175],

ii) the visualisation of feature importances in deep convolutional neural networks [163, 244, 292],

iii) information theory-based investigations of the training process [1, 222, 235, 260], and

iv) statistical analyses of weight interactions [266].

Furthermore, Raghu et al. [198] use their *expressivity* measure to explore the benefits of batch normalisation and to define a novel regularisation method. However, the authors underline that providing informative insights in combination with theoretical generality remains to be an important challenge. In the following sections, we introduce a method that aims to explain the inner workings of neural networks considering both these aspects.

We propose neural persistence (NP), a novel measure for characterising a network's structural complexity. Neural persistence adopts a novel perspective that integrates the network's connectivity and weights without interrogating it using input data. We build our method using topological data analysis, a set of techniques from algebraic topology. In the context of machine learning, TDA showed promising results in feature extraction [109, 110, 111, 112], to learn hidden representations that respect topological properties of the input data [177], or to characterise the decision boundary of neural networks [21, 99]. Lastly, Khrulkov and Oseledets [130] used TDA to describe the complexity of GAN sample spaces. In a broader scope, this work complements existing graph signal processing approaches [132, 135, 197] by investigating network topology at multiple scales using persistent homology. More specifically, in the following sections, we rephrase fully-connected neural networks into the language of algebraic topology and assess the structural complexity of i) individual layers, and ii) the entire network. Finally, we demonstrate the utility of neural persistence by developing an NP-based early stopping criterion that does not necessitate a separate validation set, making our method one of the few approaches [73, 166] that frees data to be used for training.

4.2 Topological Data Analysis and Persistent Homology

Topological data analysis is a growing field that utilises the mathematical framework of *algebraic topology* to provide computational tools for analysing complex data. Over the past few years, TDA has seen wide adaption in the machine learning community. More specifically, TDA has been used in various ways along the machine learning pipeline ranging from feature extraction for subsequent use in ML models [205], to methods that use TDA to analyse the model [297], to works that use TDA to influence model training itself [208]. For a thorough review of the field of topological machine learning, please refer to the overview article by Hensel et al. [106].

Similar to many works in topological machine learning, we use PH, a theory developed to understand high-dimensional manifolds [74, 75] as the foundation of the method we develop in the following sections. PH has been used to characterise graphs [206, 238], to find relevant features in unstructured data data [162], and to analyse image manifolds [44]. In this section, we will introduce the key concepts our method builds upon; we will follow the definitions provided by Rieck [204] and introduce *abstract* simplicial complexes and their building blocks. This view differs from the *geometrical* one (e.g. see Edelsbrunner and Harer [74, Part III] or Bredon [37, pp. 245-250]) and will allow us to use them for our computational purposes. A simplicial complex is a data structure used to represent topological spaces in a manner that is amenable to computational efforts. The constituent building blocks of a

simplicial complex are called simplices and can take the form of points, edges, triangles, and their higher-dimensional equivalents [204].

Definition 4.1 (Abstract Simplex)**.** Any subset of cardinality $k+1$ of a family of sets is called a k-simplex. In the context of this thesis, we can think of a 0-simplex as node, a 1-simplex as edge, and a 2-simplex as triangular face [204, Definition 3.2]. A geometric representation of three k-simplices is shown in Figure 4.1a.

Abstract Simplicial Complexes A simplicial complex K is a high-dimensional generalisation of a graph commonly used for the description of objects such as manifolds and defined as follows.

Definition 4.2 (Abstract Simplicial Complex [204, Definition 3.3])**.** Given a family of sets K with a collection of subsets L, K is called an *abstract simplicial complex* if:

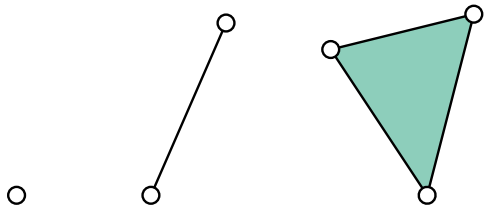
1. $\{v\} \in L$ for all $v \in$ K. The sets of the form $\{v\}$ are the *vertices* of the simplicial complex.

2. If $\sigma \in L$ and $\tau \subseteq \sigma$, then $\tau \in L$. We will refer to τ as *face* of σ.

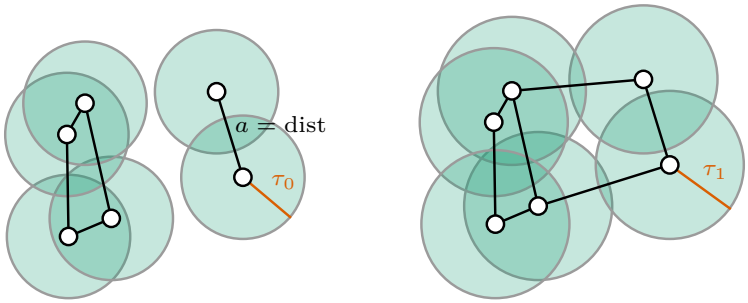
The first property requires that the complex contains all 0-simplices (i.e. vertices); the second one ensures that the simplices only intersect along shared boundaries. In the following sections, we will investigate the temporal behaviour of graphs, which, as it turns out, are represented by the so-called 1-skeleton of a simplicial complex.

Definition 4.3 (k-skeleton of a simplicial complex)**.** Given a simplicial complex K, the subcomplex containing all simplices with dimension $\leq$ k is called the k-skeleton of K. The 1-skeleton contains only vertices and edges and is thus a graph [204, Definition 3.4].

One notion to describe the connectivity of K is called simplicial homology. It uses matrix reduction algorithms [180] to derive a description of the topology of K. More precisely, it derives a *set* of such descriptors called homology groups. These features of dimension d include connected components ($d = 0$), tunnels ($d = 1$), and voids ($d = 2$) and are colloquially referred to as holes. The so-called d^{th} Betti number β_d counts the number of d-dimensional features (i.e. it summarises the information of the d^{th} homology group). For instance, a circle has the Betti numbers $(\beta_0, \beta_1) = (1, 1)$, i.e. a single connected component and a single tunnel. In contrast, a *filled* circle has Betti numbers $(1, 0)$, i.e. a single connected component but no tunnel.



(a) Examples of k-simplices: Vertex (k=0), edge (k=1), and triangle (k=2).



(b) Snapshot from a filtration process. Vertices are connected if their distances reach a specific threshold. In the left step, two connected components are present ($\beta_0 = 2$), one step later, only one is left ($\beta_0 = 1$).

Figure 4.1: Example of k-simplices and two steps from a filtration process.

Persistent Homology When it comes to the analysis of real-world data (e.g. graph data sets or neural networks), Betti numbers are limited in their use as they are too coarse and unstable to expressively describe an object. A more fine-grained method to trace the topology of an object is *persistent homology* (PH). It provides descriptions of a given object in terms of scales over which features in a homology group *persist*. More specifically, given a simplicial complex whose simplices are endowed with m function values $a_1 < a_2 < \cdots < a_m$, we can build a nested sequence of simplicial complexes

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \cdots \subseteq \mathrm{K}_m = \mathrm{K}, \tag{4.1}$$

where $\mathrm{K}_i = K(a_i)$ is the sublevel set of K at a_i, i.e. the set of simplices in K whose function value is less than or equal to a_i. This sequence is called filtration and it is common to think of the function values a_i as scale parameters that allow us to investigate topological properties of a single object at different scales. Due to the order of function values, Equation 4.1 describes the growth process of K. Figure 4.1b depicts two steps of this growth process. By endowing each potential edge with the distance between the connected vertices, we include all edges in the simplicial complex at step i if their function value is less than or equal to $2\tau_i$. The balls drawn around each vertex help to build an intuition of how the filtration comes

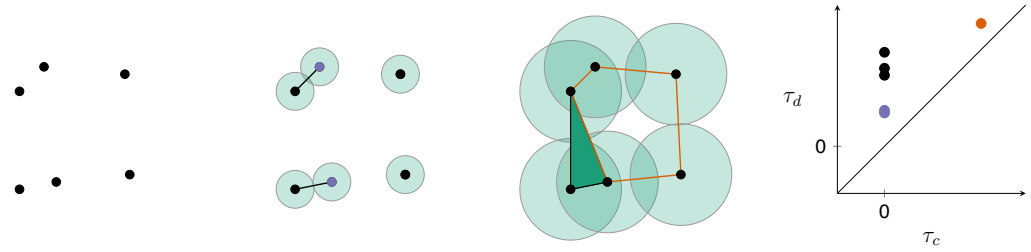


Figure 4.2: Illustration of the filtration process of a point-cloud and the resulting persistence diagram (PD). All *connected components* (i.e. 0-dimensional topological features) are created at $\tau_c = 0$ (first panel). At the threshold shown in the second panel, both purple points are merged with their closest neighbouring connected components. The thresholds at which these merges occurred are visualised in the persistence diagram in purple. The last *visualised* filtration step shows the appearance ($\tau_c > 0$) of a 1-dimensional topological feature (i.e. a tunnel) highlighted in orange. As soon as all points are connected (not shown), this tunnel vanishes and its destruction threshold is represented by the orange point in the persistence diagram. Note that we use one PD to show both 0- and 1-dimensional topological features.

into place: We start with $\tau_0 = 0$ and increase[1] it until all vertices are connected. There are two topological events that can occur during this growth of K. First, a new connected component may be created when a vertex is added, and second, two connected components may merge into one, thereby destroying one of them. These changes are tracked by persistent homology, storing creation and destruction of a topological feature as a tuple of the form $(a_i, a_j) \in \mathbb{R}^2$ for $i \leq j$ in a so-called persistence diagram (PD) as shown in Figure 4.2. We denote the collection of all tuples that correspond to d-dimensional topological features (i.e. the d^{th} PD) by $\mathcal{D}_d$. Each point in $\mathcal{D}_d$ can be summarised by a quantity called *persistence*: For a given point (x, y), we have $\text{pers}(x, y) = |x - y|$. Persistence enables us to rank topological features by how long they "live" during the filtration process and provides a measure of feature relevance. A topological feature with small persistence is destroyed briefly after its construction, a small scale event which is considered noise [75]. In contrast, a feature that persists over multiple filtration thresholds is considered a topologically relevant feature. The sum over all persistence values of a PD summarises the activity of topological features, where high values correspond to a more structured input, whereas low values indicate a high level of (topological) noise.

[1] In the development of our method, however, we will start with a fully connected graph and continuously *decrease* the distance threshold.

4.3 Persistent Homology and Neural Network Complexity

The following sections are based on the hypothesis that artificial neural networks undergoing training become increasingly structured. According to the notion that persistence, as introduced above, is a measure of topological structure, we hypothesise that it can be used to quantify the impact training has on the network's topology. We propose neural persistence (NP), a new measure that captures the structural complexity of fully-connected neural networks. Our method uses persistent homology to derive a scalar value that describes the expressiveness of a network making use of both weight information and network structure. Figure 4.3 illustrates the filtration process a neural network undergoes when computing NP.

4.3.1 Neural Persistence (NP)

Let $\mathcal{W}$ be the weights of a feedforward neural network whose neurons are connected by edges E. Due to the fact that $\mathcal{W}$ changes during training, a map $\varphi\colon E \to \mathcal{W}$ is required that maps an edge to a weight. If the activation function is fixed, all connections can be seen as a *stratified graph*.

Definition 4.4 (Stratified graph and network layers)**.** A multipartite graph $G = (V, E)$ is called stratified if its vertices are composed of the disjoint union of individual vertex sets $V = V_0 \sqcup V_1 \sqcup \ldots$, such that if $u \in V_i$, $v \in V_j$, and $(u, v) \in E$, we have $j = i + 1$. Hence, edges are exclusively allowed between vertex sets that are adjacent. For $k \in \mathbb{N}$, we refer to the unique subgraph $G_k := (V_k \sqcup V_{k+1}, E_k := E \cap \{V_k \times V_{k+1}\})$ as the k^{th} *layer* of the stratified graph.

This means that, once all weights are sorted, we can compute persistent homology of each G_k and G using their filtration. Such an approach is reminiscent of topology-based network analyses, in which weights commonly describe node similarity or closeness [45, 113]. In contrast to these network analyses, the weights of the neural networks we are interested in are constantly changing and may result in unbounded values. This observation necessitates a novel filtration procedure. It is common to base a filtration on the Euclidean distance between samples when using PH [41]. As illustrated in Figure 4.1b, a filtration starts by connecting close points first and, with a growing threshold, continuously creates edges between nodes that are increasingly distant from each other. In the following paragraph, we will develop the filtration NP is based on.

Filtration Let $\mathcal{W}$ be the set of weights of an individual training step. Furthermore, we have $w_{\text{max}} := \max_{w \in \mathcal{W}} |w|$. Before defining our filtration, we normalise these weights as

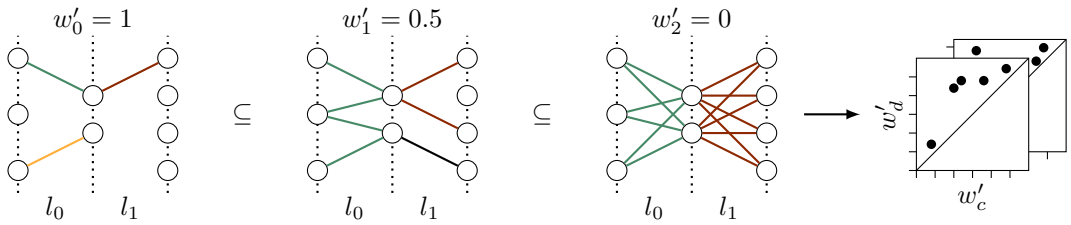


Figure 4.3: Illustration of neural persistence computation given a network with layers l_0 and l_1. Colours indicate individual connected components. Connected components are either merged or created during the filtration process when their weights are greater than or equal to the threshold w'_i. With decreasing w'_i, the connectivity of the network increases. The thresholds at which a topological feature gets created or destroyed are summarised in a persistence diagram. For each layer, one persistence diagram exists, which is subsequently used to compute neural persistence according to Equation 4.2.

follows. We have $\mathcal{W}' := \{|w|/w_{\max} \mid w \in \mathcal{W}\}$ which is indexed in non-ascending order, s.t. $1 = w'_0 \geq w'_1 \geq \cdots \geq 0$. Furthermore, let $\varphi'(u, v) \in \mathcal{W}'$ be a function that assigns the transformed weight to a given edge. We can now define the following filtration. Let G_k be the k^{th} layer of the network, then $G_k^{(0)} \subseteq G_k^{(1)} \subseteq \ldots$, where $G_k^{(i)} := (V_k \sqcup V_{k+1}, \{(u, v) \mid (u, v) \in E_k \wedge \varphi'(u, v) \geq w'_i\})$. This filtration was modified in a way such that it fits well into the framework of neural networks. In contrast to other graph-structured data (e.g. social networks), it is crucial to appreciate the relevance of absolute edge weights. In artificial neural networks, high absolute values go hand in hand with increased impact on subsequent layers. In the proposed filtration, weak connections ($|w| \approx 0$) remain close to 0 and the overall strength of a connection is maintained. Moreover, due to the normalisation, $w' \in [0, 1]$ holds, making the neural network scale-invariant making different neural networks comparable.

Persistence Diagrams After defining our novel filtration, we can compute PH for each layer G_k. For this, let us first emphasize that we capture "merely" zero-dimensional topological information. This is due to the fact that the filtration as defined above includes only nodes and edges, i.e. 0-simplices and 1-simplices. Hence, we are able to measure creation and destruction of connected components at different scales. We would like to stress that including higher-dimensional information is possible [206], however the advantages of focusing on zero-dimensional information are as follows:

i) it is possible to interpret resulting values since they allow to infer clustering information about the network at different weight thresholds,

ii) previous research [112, 207] indicates that large amounts of information are captured by zero-dimensional topological information, and

iii) PH calculations are computationally highly efficient (see below).

The structure of the persistence diagrams generated by our filtration is particular: in the beginning, all vertices exist, which means they are part of $G_k^{(0)}$ for each k. This is due to the fact that we only sort *edges*. Consequently, their assignment of a weight of 1 results in $|V_k \times V_{k+1}|$ connected components and persistence diagrams whose entries are of the form $(1, x)$, with $x \in \mathcal{W}'$. The latter implies that respective tuples are always *below* the diagonal, a property that is reminiscent of so-called superlevel set filtrations [41, 53]. We are now in the position to define neural persistence by using the p-norm of a persistence diagram (PD) as introduced by Cohen-Steiner et al. [54]:

Definition 4.5 (Neural persistence)**.** Given the k^{th} layer of a neural network, its neural persistence is denoted by $\mathrm{NP}(G_k)$. It is defined as the p-norm of $\mathcal{D}_k$ (its persistence diagram) constructed in the same way as introduced above, i.e.

$$\mathrm{NP}(G_k) := \|\mathcal{D}_k\|_p := \Big(\sum_{(c,d)\in\mathcal{D}_k} \mathrm{pers}(c,d)^p \Big)^{\frac{1}{p}}, \tag{4.2}$$

which (for $p = 2$) summarises the Euclidean distances of points in $\mathcal{D}_k$ to the diagonal.

It has been shown [54] that the p-norm summarises topological features captured in a persistence diagram in a stable manner. Under the assumption that `NP` is an expressive and meaningful summary of a network's structural complexity, we expect it to correlate (to a certain degree) with the number of trained epochs. This requirement and other properties will be evaluated Section 4.3.3.

Algorithm 4 outlines the calculation of `NP`[2]. As touched upon before, the filtration in line 4 requires that all n network weights are sorted which has a computational complexity of $\mathcal{O}(n \log n)$. The persistent homology calculation in line 5 can be performed by utilising a union–find data structure [75]. The computational complexity of the involved computations is $\mathcal{O}(n \cdot \alpha(n))$, where $\alpha(\cdot)$ is the slow-growing inverse of the Ackermann function [55, Chapter 22].

[2] We published experiments and our implementation at `https://github.com/BorgwardtLab/Neural-Persistence`.

Algorithm 4 Neural persistence calculation

Input: Neural network with l layers and weights $\mathcal{W}$

1: $w_{\max} \leftarrow \max_{w \in \mathcal{W}} |w|$ // Determine largest absolute weight
2: $\mathcal{W}' \leftarrow \{|w|/w_{\max} \mid w \in \mathcal{W}\}$ // Normalise weights for filtration
3: **for** $k \in \{0, \ldots, l-1\}$ **do**
4: $\quad F_k \leftarrow G_k^{(0)} \subseteq G_k^{(1)} \subseteq \ldots$ // Compute filtration of k^{th} layer
5: $\quad \mathcal{D}_k \leftarrow \textsc{PersistentHomology}(F_k)$ // Calculate persistence diagram
6: **end for**
7: **return** $\{\|\mathcal{D}_0\|_p, \ldots, \|\mathcal{D}_{l-1}\|_p\}$ // Calculate neural persistence for each layer

4.3.2 Properties of Neural Persistence

In this section, we describe properties of NP that allow us to use it as a measure to compare networks with different architectures. We start by deriving lower and upper *bounds* for NP of an individual layer.

Theorem 4.1. We follow Definition 4.4 and let G_k be the k^{th} layer of a feedforward neural network. Additionally, let $\varphi_k \colon E_k \to \mathcal{W}'$ assign all individual edges of G_k a normalised weight. When we use the filtration outlined in Section 4.3.1 for our persistent homology computation, $\mathrm{NP}(G_k)$ satisfies

$$0 \leq \mathrm{NP}(G_k) \leq \left(\max_{e \in E_k} \varphi_k(e) - \min_{e \in E_k} \varphi_k(e) \right) (|V_k \times V_{k+1}| - 1)^{\frac{1}{p}}, \tag{4.3}$$

where the number of neurons of G_k is equivalent to the cardinality of the vertex set $|V_k \times V_{k+1}|$.

Proof. We prove Theorem 4.1 in a constructive manner by laying out how these bounds can be derived. Beginning with the lower bound, we denote a fully-connected layer with $|V_k|$ vertices as G_k^-. Moreover, let $\varphi_k(e) := \theta$ be a function that assigns each edge e the constant value of $\theta \in [0, 1]$. In this case, our filtration degrades to a function that orders vertices and edges lexicographically resulting in a PD exclusively consisting of tuples of the form (θ, θ). As these entries lie on the diagram's diagonal, we have $\mathrm{NP}(G_k^-) = 0$. With respect to the upper bound, we will define G_k^+ to be a layer containing at least 3 vertices ($|V_k| \geq 3$). Additionally, we have $a, b \in [0, 1]$ with $a < b$. We will make a random selection of one edge e' and define its weight function to be $\varphi(e') := b$. All other edges will be assigned the weights $\varphi(e) := a$.

In the construction of the filtration, adding the edge e' will lead to the tuple (b, b), the other pairs, however, will result in the tuples (b, a). Therefore, we have

$$\begin{aligned} \mathrm{NP}(G_k^+) &= \Big(\mathrm{pers}(b,b)^p + (|V_k| - 1) \cdot \mathrm{pers}(b,a)^p \Big)^{\frac{1}{p}} = (b-a) \cdot (|V_k| - 1)^{\frac{1}{p}} && (4.4) \\ &= \left(\max_{e \in E_k} \varphi(e) - \min_{e \in E_k} \varphi(e) \right) (|V_k| - 1)^{\frac{1}{p}}, && (4.5) \end{aligned}$$

and illustrated the realisation of the upper bound. Let us now consider the following perturbed weight function to lay out why it is not possible for this term to be surpassed by $\mathrm{NP}(G)$ for any G:

$$\widetilde{\varphi}(e) := \varphi(e) + \epsilon \in [0, 1] \tag{4.6}$$

As the difference $\max \varphi(e) - \min \varphi(e)$ maximises $b - a$ in Equation 4.4, this perturbation cannot increase neural persistence. ■

A normalising factor for a layer's neural persistence can now be derived by using the upper bound of Theorem 4.1. This allows us not only to use NP to compare individual layers but also whole networks from different architectures of varying sizes:

Definition 4.6 (Normalised neural persistence)**.** Given a layer G_k, we have

$$\widetilde{\mathrm{NP}}(G_k) := \frac{\mathrm{NP}(G_k)}{\mathrm{NP}(G_k^+)}. \tag{4.7}$$

Using the normalised formulation of NP makes individual layers comparable and allows us to compute a network's *overall* neural persistence as shown in Definition 4.7. While it is possible to apply a single filtration to the neural network as a whole, the approach outlined above prevents that layers with weights of different scales skew the construction of its PD and hence the computation of NP.

Definition 4.7 (Mean normalised neural persistence)**.** Following Definition 4.4, we consider a feedforward neural network as stratified graph G. Its *mean normalised neural persistence* is defined as the sum of all normalised NP values per layer, i.e.

$$\overline{\mathrm{NP}}(G) := \frac{1}{l} \cdot \sum_{k=0}^{l-1} \widetilde{\mathrm{NP}}(G_k). \tag{4.8}$$

While Theorem 4.1 gives a theoretical lower and upper bound in the general setting, we can obtain empirical bounds considering the tuples that result from the computation of a persistence diagram.

In addition to the theoretical bounds derived in Theorem 4.1, we can compute empirical bounds based on the tuples of a PD. The filtration introduced at the beginning of this section results in PDs containing creation and destruction points of the form $(1, w_i)$, where each $w_i \in [0, 1]$ represents a normalised weight. From this particular structure, we can derive empirical bounds that do not necessitate the computation of NP itself. A theoretical insight leading to a more efficient implementation of our measure.

Theorem 4.2. Following Theorem 4.1, let G_k be th k^{th} layer of a feedforward neural network. Moreover, let n be the number of its vertices and m the number of its edges for which we have $w_0 \leq w_2 \leq \cdots \leq w_{m-1}$. Then the following holds:

$$\|\mathbb{1} - \mathbf{w}_{\text{max}}\|_p \leq \text{NP}(G_k) \leq \|\mathbb{1} - \mathbf{w}_{\text{min}}\|_p, \tag{4.9}$$

where the vectors $\mathbf{w}_{\text{min}} = (w_0, w_2, \ldots, w_{n-1})^T$ and $\mathbf{w}_{\text{max}} = (w_{m-1}, w_{m-2}, \ldots, w_{m-n})^T$ contain the n smallest and n largest weights, respectively.

Proof. The filtration introduced in Section 4.3.1 can be seen as a constrained subset selection problem, where we are given m weights from which we select n. Hence, $\text{NP}(G_k)$ depends only on the weights that are being *selected* and appear in $\mathcal{D}_k$ as points of the form $(1, w_i)$. Let us denote $\widetilde{\mathbf{w}}$ as a vector containing the selected weights and reformulate neural persistence as $\text{NP}(G_k) = \|\mathbb{1} - \widetilde{\mathbf{w}}\|_p$. Moreover, for $\widetilde{\mathbf{w}}$ it holds that $\|\mathbf{w}_{\text{min}}\|_p \leq \|\widetilde{\mathbf{w}}\|_p \leq \|\mathbf{w}_{\text{max}}\|_p$. From the fact that our filtration only contains non-negative weights we thereby have

$$\|\mathbb{1} - \mathbf{w}_{\text{max}}\|_p \leq \text{NP}(G_k) \leq \|\mathbb{1} - \mathbf{w}_{\text{min}}\|_p, \tag{4.10}$$

and the claim follows. ■

Neural Persistence and Complexity Regimes In this section, we will investigate to which degree NP can capture the structural complexity of converging and diverging perceptrons. To do so, a perceptron is trained on the MNIST data set [145]. In this setting, our measure only uses the weight matrix of a perceptron, which permits us to compare its NP to the NP of random weight matrices, drawn from different distributions. Furthermore, we can contrast neural persistence of *trained* networks with NP of their initial state. Different settings of this experiment are depicted in Figure 4.4. We illustrate neural persistence values (dots) and its respective bounds (crosses) for different types of networks and matrices. In the centre of the plot, we show neural persistence values of random Gaussian matrices in red and NP values of perceptrons that diverged during training in yellow. We can observe that in terms of their neural persistence, both matrix "types" cannot be distinguished. That being

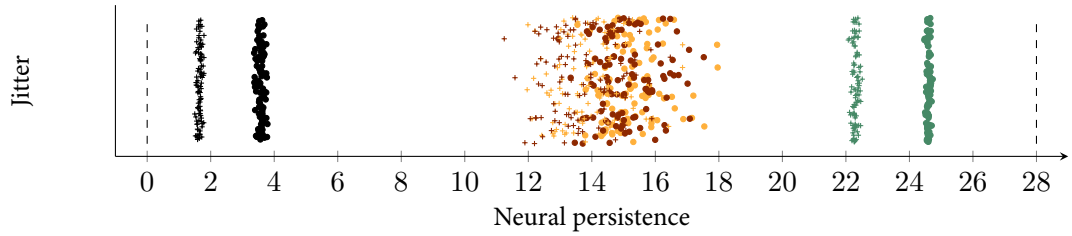


Figure 4.4: In green, we show NP values of perceptrons that are trained; in yellow diverging ones. Red dots indicate neural persistence values of random Gaussian matrices and black NP values of random uniform matrices. For each category, we performed 100; the lower bound from Theorem 4.2 is depicted as crosses while dots show actual neural persistence values. Dashed lines illustrate the bounds derived in Theorem 4.1.

said, perceptrons that converge during training (shown in green) exhibit significantly higher NP values. Black dots and crosses indicate lower bound and neural persistence of random uniform matrices which display the lowest NP values. This observation confirms the intuition that Gaussian matrices contain only few neurons with large (absolute) weights. Note that the distribution of most weights are heavily right-tailed resulting in an empirical upper bound not as tight as tight as the lower one. Hence, we do not show this upper bound.

Neural Persistence and Graph-Theoretical Measures Sizemore et al. [238] showed that for the characterisation of small random networks, PH can outperform graph-theoretical structure/complexity measures such as the shortest path length and clustering coefficient. As the above and following experiments show, this also holds for deep feedforward neural networks. To elucidate this observation, we used the MNIST data set to train perceptrons in two settings. First, a "successful" learning rate of $\eta = 0.5$ was chosen leading to trained networks with test accuracies of ≈ 0.91. Second, we intentionally "sabotaged" the training by selecting a low learning rate of $\eta = 1 \times 10^{-5}$, preventing the training process from converging. The accuracies reached by these diverging networks range from 0.38 to 0.65. From a machine learning perspective, these networks belong to two different classes, between which a useful complexity measure should differentiate. Figure 4.5 (top) clearly indicates that clustering coefficient is not able to quantify these differences. In contrast, our method (bottom) can distinguish both classes well. Not only do we observe two distinct neural persistence regimes but also a notable smaller variance for trained networks.

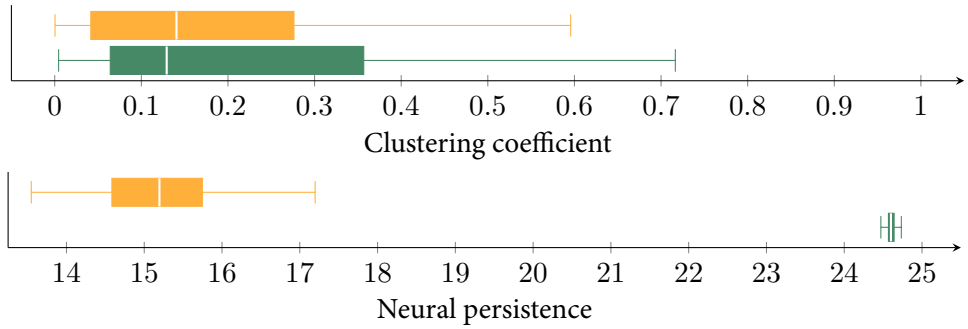


Figure 4.5: Distribution of network complexity measures computed for two types of feedforward networks. Properly trained networks are shown in green ($\eta = 0.5$), diverging ones in yellow ($\eta = 1 \times 10^{-5}$). The clustering coefficient (top) is a traditional graph measure that fails to detect the structural differences of both neural network classes. The plot on the bottom shows that neural persistence (NP) for trained networks follows a different distribution than NP for diverging networks.

4.3.3 Experiments

In the following, we will demonstrate the relevance and utility of NP as a meaningful descriptor of the structural complexity of deep artificial neural networks. We will first examine how standard regularisation techniques such as dropout and batch normalisation affect our measure. This investigation is followed by the development of an early stopping criterion that uses neural persistence to determine whether to stop training. As NP only measures network complexity, it is different from the traditional approach where a validation data set is required. Across experiments, we used a variety of architectures with *rectified linear unit* (ReLU) activation functions. We follow a notation in which the size of a *hidden* layer (i.e. the number of neurons) is denoted in brackets. Unless noted otherwise, all networks are fitted using the Adam optimiser [133] and parameters are tuned via cross-validation. Table 4.1 provides a complete list of hyperparameters and experimental details of all executed experiments.

Table 4.1: Parameters and hyperparameters for the experiments on best practices. Batch normalisation and dropout were always applied after the first hidden layer. We trained all networks with the *ReLU* activation function.

Data set	# Runs	# Epochs	Architecture	Optimiser	Batch Size	Hyperparameters
MNIST	50	40	$[650, 650]$	Adam	32	$\eta = 0.0003$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$
						$\eta = 0.0003$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, Batch Normalisation
						$\eta = 0.0003$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, Dropout 50%

Table 4.2: Parameters and hyperparameters for the experiment on early stopping. Throughout the networks, *ReLU* was the activation function of choice.

Data set	# Runs	# Epochs	Architecture	Optimiser	Batch Size	Hyperparameters
(Fashion-)MNIST	100	10	Perceptron	Minibatch SGD	100	$\eta = 0.5$
			$[50, 50, 20]$			
		40	$[300, 100]$	Adam	32	$\eta = 0.0003$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$
			$[20, 20, 20]$			
CIFAR-10	10	80	$[800, 300, 800]$	Adam	128	$\eta = 0.0003$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$
IMDB	5	25	$[128, 64, 16]$	Adam	128	$\eta = 1 \times 10^{-5}$ $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$

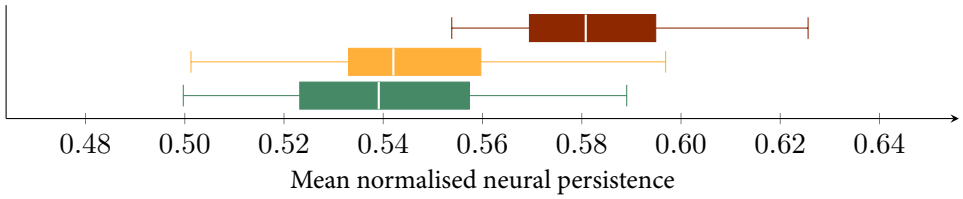


Figure 4.6: Mean normalised neural persistence for networks trained with batch normalisation (yellow) and dropout (red). We show unmodified networks in green. We trained each setting 50 times, and set the dropout rate to $p = 0.5$.

4.3.3.1 Neural Persistence and Deep Learning Best Practices

We compare the mean normalised neural persistence of two-layer neural networks (with $[650, 650]$ architecture) for which either batch normalisation [118] or dropout [245] are applied. We use the MNIST data set for training and depict the results in Figure 4.6. Compared to "off-the-shelf" networks (green), neural nets trained using deep learning best practices result in higher NP values. Dropout, in particular, seems to affect normalised neural persistence more than batch normalisation, a trend which is also observed in the test set accuracy. Considering dropout as ensemble learning as done by Hara et al. [102], these results line up with our expectations. More specifically, dropout leads to the independent training of individual network parts, which results in a higher level of per-layer redundancy, changing the network's structural complexity. Overall, these results indicate that at least for fixed architectures, techniques that increase neural persistence throughout training may be of specific importance.

4.3.3.2 Validation-Free Early Stopping Based on Neural Persistence

In this section, we will use NP as *early stopping* criterion that helps preventing overfitting by considering neural network structure alone. To test whether our measure can be used this way, we proceed as follows. We continuously monitor mean normalised neural persistence during network training and stop the optimisation if NP plateaus. More specifically, if there is no increase in NP of more than Δ_{min} for g epochs, the training process is halted. Algorithm 5 outlines this patience-based procedure. When we exchange neural persistence for validation loss, this algorithm is the most commonly used early stopping methods in the training of neural networks [19, 51]. We investigate the efficacy of NP as early stopping criterion and compare it to the standard validation loss approach. More concretely, in our experiments, we train networks for no more than G epochs and specify a $G \times G$ grid containing g, the patience parameter, and b, the burn-in rate. We measure both parameters in number of epochs. The burn-in rate defines the epoch at which a criterion (i.e. NP or validation loss) is started to be

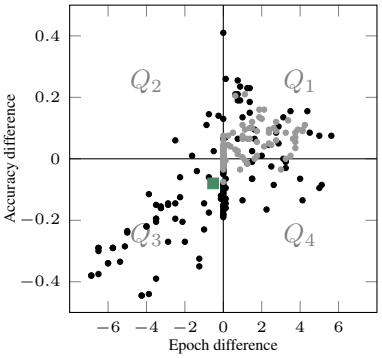
Algorithm 5 Using mean normalised neural persistence as stopping criterion

Input: Weighted neural network $\mathcal{N}$, patience g, Δ_{min}

$P \leftarrow 0, G \leftarrow 0$ // Initialize highest observed value and patience counter
procedure EARLYSTOPPING($\mathcal{N}$, g, Δ_{min}) // Callback that monitors training at every epoch
    $P' \leftarrow \overline{\text{NP}}(\mathcal{N})$
    if $P' > P + \Delta_{\text{min}}$ **then** // Update mean normalised neural persistence and reset counter
        $P \leftarrow P', G \leftarrow 0$
    else// Update patience counter
        $G \leftarrow G + 1$
    end if
    if $G \geq g$ **then** // Patience criterion has been triggered
        return P // Stop training and return highest observed value
    end if
end procedure

used to monitor training. To keep the experiment comparable and scale-invariant, Δ_{min} is set to zero, preventing non-zero values to favour one method over the other. We perform 100 training runs per grid cell with identical architectures per data set for each of four data sets. Both validation loss and mean normalised `NP` are computed every quarter epoch. We simulate both measures' stopping behaviour for every combination of g and b and summarise their predictive performance by computing the median test accuracy over all runs. Similarly, we record the median stopping epoch to assess how late/early a criterion is triggered. For runs in which a criterion was never triggered, test accuracy and number of training epochs were recorded as soon as training was finished.

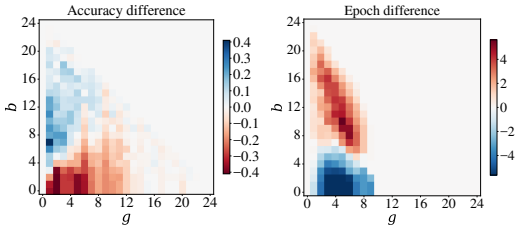
In the resulting scatterplot, each point corresponds to a unique parameter configuration and shows the absolute test accuracy difference (in percent) and the difference in epochs after which training was stopped. To make the plot more accessible, we split it into four quadrants: If a parameter combination results in a scenario in which `NP`-based early stopping reaches a *higher* accuracy and stops *earlier* than the validation loss-based criterion, it appears in quadrant Q_2. We show configurations for which b or g are more than half the number of total training epochs in grey as they represent uncommon parameter settings (i.e. typically, both parameters are comparatively small). Finally, we summarise our measure's performance by computing the barycentre over all combinations (green square).



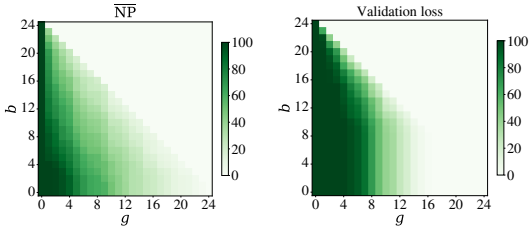
(a) Fashion-MNIST

Data set	Barycentre	Final test accuracy
Fashion-MNIST	$(-0.53, -0.08)$	86.72 ± 0.43
MNIST	$(+0.17, -0.06)$	96.16 ± 0.24
CIFAR-10	$(-1.33, -1.13)$	52.19 ± 3.40
IMDB	$(-1.68, +0.07)$	87.35 ± 0.03

(b) Summary



(c) Accuracy and epoch differences

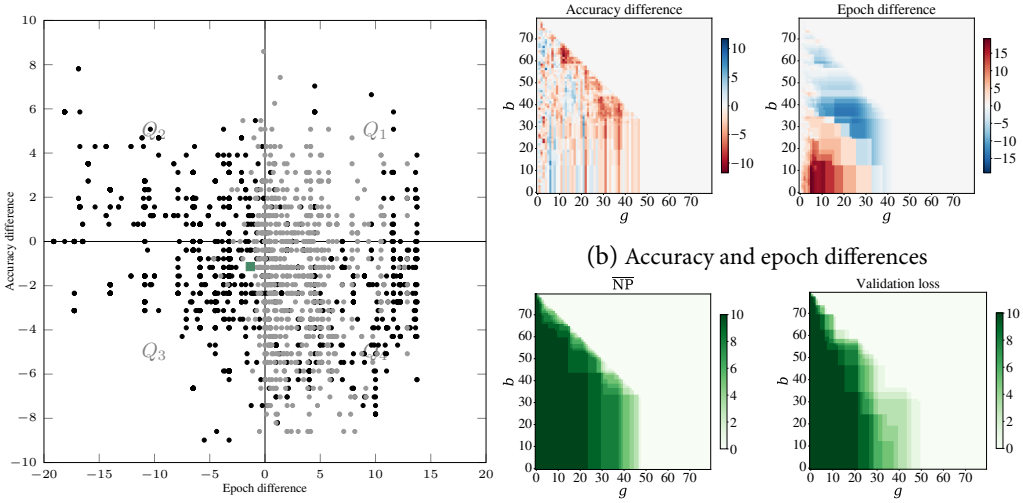


(d) Number of triggers

Figure 4.7: Differences (validation loss v.s. neural persistence) in number of epochs trained and test accuracy for all combinations of g and b on the Fashion-MNIST data set. Table 4.7b summarises the results of all four data sets. To provide a holistic evaluation, final accuracy values are considered even if no early stopping criterion was triggered.

We compare our approach with the validation loss-based criterion on the Fashion-MNIST [284] data set in Figure 4.7a and observe that almost all configurations are in Q_3 or in Q_2, i.e those quadrants in which our criterion stops earlier. The barycentre is located at $(-0.53, -0.08)$, indicating that out of all 625 parameter combinations, on average, the NP-based stopping leads essentially to the same accuracy (0.08%) as validation loss while stopping half an epoch before. A more detailed depiction of epoch and accuracy differences is visible in Figure 4.7c; each heatmap cell corresponds to a specific parameter combination of g and b. Red, white, and blue represent configurations for which we get *lower, equal, or higher* accuracy, respectively, than using validation loss with the same parameters. Likewise, blue/red values in the heatmap that visualises epoch differences indicate that our criterion triggered earlier/later than validation loss. With burn-in periods of $b \leq 8$, our criterion stops on average 0.62 epochs earlier suffering only a small decrease in predictive accuracy (0.06%). Finally, Figure 4.7d shows the frequency at which each measure stopped the training process. A consistent stopping criterion would lead to a (dark) green triangle. To summarise Figure 4.7d, we see that on Fashion-MNIST, our approach stops not as frequently overall, but for more parameter configuration than the validation loss-based early stopping.

We performed the same analysis on the CIFAR-10 data set [138] and show its results in Figure 4.8. Overall, we make the observation that both criteria stop less consistently on this data



(a) CIFAR-10

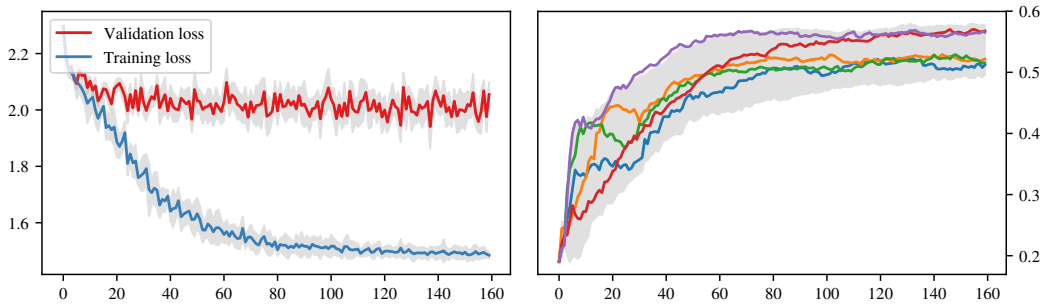
(b) Accuracy and epoch differences

(c) Number of triggers

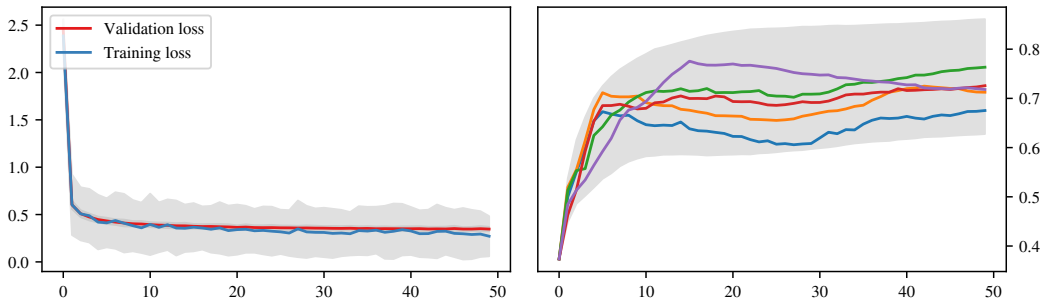
Figure 4.8: Early stopping behaviour for the CIFAR-10 data set.

set compared to the results on Fashion-MNIST. More specifically, there are configurations for which NP stops earlier and leads to an accuracy increase of almost 10% but also combinations for which our approach does not stop earlier or has to train much longer. Concerning reliability, however, we show in Figure 4.8c that our measure triggers for more combinations compared to validation loss. In addition, from the scatterplot in Figure 4.8a, we can see that most black dots (i.e. practical parameter combinations) are located in Q_3 and Q_2. Regarding settings in which we reach higher accuracies and train longer (i.e. Q_1), we notice that they are characterised by a small burn-in rate (b) and high patience (e.g. $g \approx 40$), or vice versa.

We hypothesise this may be due to the fact that for CIFAR-10, fully connected neural networks do not reliably converge. To demonstrate this, we show mean normalised persistence and loss curves in Figure 4.9. With respect to validation and training loss, coloured curves illustrate their mean over 5 training runs. Additionally, we show mean normalised neural persistence values in colour for all individual runs. Grey envelopes depict standard deviations. Note that we only show the first 50 epochs as they represent practical/useful early stopping settings. For the Fashion-MNIST data set, we see obvious change points at which $\overline{\text{NP}}$ can be used to halt the training process. In contrast, for the CIFAR-10 data set, it is harder to find well-defined maxima in some training runs, which exacerbates deriving a general stopping criterion without parameter fine tuning. Therefore, we hypothesise that it is not possible to use our measure reliably in situations where the network is not capable of generalising to the validation set.



(a) CIFAR-10



(b) Fashion-MNIST

Figure 4.9: Comparing the training behaviour in terms of training/validation loss as well as mean normalised NP on the CIFAR-10 and Fashion-MNIST data sets. Losses are averaged over five runs for each which we show the neural persistence trajectory. Grey envelopes show standard deviations.

4.4 Conclusion

In this section, we introduced neural persistence (NP), a new complexity measure for deep neural networks. Inspired by advances in the field of computational topology, neural persistence is not only computationally efficient but also enjoys a strong theoretical foundation. We demonstrated its general applicability and that it captures useful topological information pertaining to a network's generalisation performance. We also illustrated that our measure can identify networks trained by using deep learning best practices like normalisation and dropout. Additionally, we showed that when used as criterion for early stopping, NP results in networks that reach competitive prediction performance without necessitating a distinct validation set. Freeing data for training is particularly desirable to use deep learning in settings in which limited data is available. That being said, neural persistence is a coarse-grained measure that quantifies large-scale changes in a network's structure that are more pronounced at the beginning of training. Smaller weight refinements that happen towards the end of the training process may be too subtle to be captured by NP. This is underlined by the observa-

tion that our early-stopping criterion leads to *comparable* predictive performance but not to an increase.

As one of the pioneering works at the intersection of machine learning and TDA, `NP` contributed to the emergence of the field of topological machine learning. In a follow-up work [205], the author of this thesis underlined the versatility and importance of TDA by developing a topological generalisation of the popular Weisfeiler-Lehman procedure [280] for graph characterisation and classification [233, 234]. Over the last years, many works in topological machine learning followed threads first touched upon by neural persistence. In particular, the thread of characterising neural networks and their generalisation capabilities using TDA was investigated extensively [52, 56, 88, 184, 279]. Other new paths pertain to the problem of learning disentangled representations [297] or topological-inspired approaches to autoencoders [177], to name a few.

5 Concluding Remarks & Outlook

In which we summarise our contributions and describe promising future research directions.

We began this thesis by defining two canonical classes of time series: real-valued time series, containing what is commonly referred to as uni- and multivariate time series, and object-valued time series, which encompass temporal changes of arbitrary objects. By focusing primarily on the biomedical context, we then identified three challenges in the analysis of real-valued time series. These include statistical obstacles arising from the search for temporal biomarkers, weaknesses in the use of subsequence-based classification algorithms, as well as a lack of accurate and non-invasive methods for cardiac risk stratification. Other computational challenges that will drive the development of new and effective time series analysis methods evolve around the utility of generative models for time series and continuous inference of discrete signals. In the second part, we turned towards object-valued time series by investigating the inner workings and learning behaviour of one of the most successful objects in machine learning: artificial neural networks. Throughout the thesis, we proposed solutions to these challenges by leveraging advances in pattern mining, kernel methods, deep learning, and topological data analysis. In this chapter, we will first summarise the solutions that were developed in this thesis to solve or mitigate the aforementioned challenges. Each summary will be followed by elaborations on future research directions that we consider to be promising. We then conclude by providing general directions for the study of biomedical time series.

Pattern Mining for Time Series

Chapter 2 introduced `S3M`, an algorithm to identify discriminating patterns that are statistically associated with a binary class label. `S3M` extends the idea of time series shapelets [288], originally developed for classification, to the realm of pattern mining. The problem of multiple hypothesis testing is mitigated by including Tarone's method [257] in the subsequence mining process. A novel contingency table pruning procedure makes `S3M` particularly efficient without losing statistical power. Applied to a sepsis data set, we demonstrated the

utility of our method by identifying septic-specific temporal patterns in a patient's heart rate, respiratory rate, and blood pressure.

In a follow-up work [117], we showed how S3M could be used as a filtering method to select a limited number of shapelets as time series feature descriptors. We demonstrated how our method could not only be used in association mapping but also in classifying complex phenotypes from time series. Lastly, S3M was extended by Gumbsch et al. [97], who enhanced the practical utility of the method by reducing the number of similar shapelets that are reported.

For future work, we envision S3M to be used for the data-driven definition of complex phenotypes such as sepsis. The current definition of sepsis [237] was developed by clinical trial experts, specialists in sepsis pathobiology, and epidemiology scholars. While clinical expertise and diligent consensus-finding processes are crucial for the development of clinical definitions, we foresee great potential in the use of computational approaches to augment current practices. The benefit of S3M over other approaches is its interpretability and statistical soundness, both valuable properties, which we think are critical for the acceptance of data-driven derivations of clinical definitions. To reach this goal, the first step would be to extend our method to multivariate time series, enabling it to detect interacting patterns from different channels. The challenge of such an extension, however, is the combinatorial explosion of possible interactions. A first way to mitigate this problem is to only consider combinations of patterns that are in close temporal proximity. Second, we must not rely on labels derived by physicians but proceed in an unsupervised fashion. This may be achieved by an iterative label-assigning approached coupled with S3M. We could start from a data set comprised of unlabelled time series and randomly assign labels in an iterative manner. In each iteration, S3M is used to mine shapelets. If any statistically associated shapelet is found, we keep the current label distribution and assign positive labels to samples from the current negative class. This way, we construct two classes of time series for which statistically associated shapelets exist. A subsequent medical interpretation of the results would be necessary to confirm the validity of the assigned labels and identified shapelets.

Subsequence Kernels for Time Series

In the beginning of Chapter 3, we propose the Wasserstein Subsequence Kernel (WTK), a kernel-based method for time series classification that makes use of concepts from optimal transport theory. We show that a simple application of the $\mathcal{R}$-convolution kernel framework employing subsequences can be meaningless when assessing the similarity of two time series. Motivated by this observation, we developed a meaningful kernel method that allows for a less rigid notion of similarity. Specifically, we used the 1-Wasserstein distance between subsequences to capture local *and* global time series characteristics concurrently. We provide

empirical evidence of the effectiveness of our method across a wide range of data sets and in a wide range of settings.

A first step in making our method widely applicable is scaling `WTK` to longer time series. One of the current computational bottlenecks arises from the number of subsequences under consideration as it defines the dimensionality of the ground distance matrix. As longer time series lead to a higher number of subsequences, it may become critical to drastically reduce the number of subsequences that represent a time series. `S3M` or its extension [97] may prove to be a valuable, data-driven subsequence selection method for this task. Another interesting research direction is to use time series representations that differ from subsequences. A fruitful advancement may be found in the use of the signature transform [131, 148], a universal nonlinearity that has been proven to be an effective feature representation for time series. While this does not entail a gain in computational complexity, it may result in increased expressive power. Moreover, to circumvent the challenge of dealing with an indefinite kernel function, a feasible approach is to use the sliced Wasserstein distance instead [136, 137, 142]. This distance is provably negative definite, which allows for the derivation of a positive definite kernel as shown by Feragen et al. [82]. This brings us to the last research direction we identified. Once provided with a positive definite kernel, we can expand `WTK`'s application beyond classification and explore areas such as dimensionality reduction using kernel principal component analysis [226] or two-sample tests using the maximum mean discrepancy (MMD) [32, 96] framework.

Ischaemia Prediction with Deep Learning

In the second section of the time series classification chapter, we developed a system based on deep learning for the identification of myocardial ischaemia from exercise stress test data. Exercise-induced myocardial ischaemia is the hallmark of coronary artery disease (CAD), the leading cause of years of life lost. The early identification of patients at risk for CAD is therefore an important medical and epidemiological task that may lead to a more targeted testing strategy. However, the gold standard to determine myocardial ischaemia, namely myocardial perfusion imaging, comes with disadvantages. It is intrusive in nature and exposes the patient to radioactive agents. A system that can predict the outcome of the imaging from easy-to-access data is therefore particularly desirable. If this system were sufficiently accurate, it could reduce the number of false positives (i.e. patients who unnecessarily received imaging) and thus spare patients from the procedure. Our system not only reduces the number of false positives *and* false negatives but also exhibits interpretability properties that help the cardiologist understand which parts of the input data contributed to the predicted risk score.

While we were able to internally validate the clinical value of our system, we envision externally validating our system's predictive performance on data from different institutions providing a multi-site evaluation. In particular, it is of interest to investigate the system's generalisation capabilities to ECG data recorded via treadmill stress testing (as compared to bicycle stress testing). In addition to bicycle testing, treadmill testing is another widespread protocol that results in a slightly higher physical load due to the fact that it is not weight-bearing. This will not change the physiological manifestation of ischaemia; however, it may provide insights into the robustness of our system with respect to different sources of noise. An available data set that can be used for this purpose was generated by Sharir et al. [232] and made available as part of the Telemetric and Holter ECG Warehouse (THEW) initiative [57]. Moreover, there are many powerful neural network architectures well suited for sequential data whose inductive biases may be beneficial for the classification task at hand. For instance, attention-based models [270] are able to learn long-term dependencies and are therefore particularly well-suited to our task (e.g. to learn long-term ST-segment changes). While the original method suffers from a computational complexity quadratic in the length of the time series, efficient approaches such as the "Reformer" [134] or "Perceiver" [119] may be useful in our setting. Moreover, to sustainably improve the standard of care, it is necessary to confirm the system's added value in a clinical study. The system should be evaluated in the following two scenarios. First, in a setting where no cardiologist is available to interpret stress test results (e.g. at a general practitioner's office), the system could help identify patients at risk that are hard to identify for non-experts. In this setting, the primary aim is to reduce the false negative rate. Second, in specialised clinics with access to an expert's judgement, we might be more interested in reducing the number of patients that are unnecessarily exposed to radioactive agents. Not only would this lead to increased patient safety but also to a reduced financial burden for the hospital. In addition to improvements in risk stratification, future work could also shed light on the interplay between machine learning systems and physicians. A prospective study investigating this may provide insights about the influence that machine-generated scores have on the cardiologist's judgement and the degree to which model interpretability can increase the trust of healthcare professionals in an algorithm.

Neural Networks as Time-Varying Graphs

Object-valued time series were the focus of Part II. We defined this class of time series as sequences of structured objects whose nature determines the analysis methods that can be utilised. The temporal change of weighted stratified graphs that represent neural networks that change during training were investigated. We developed neural persistence (`NP`), a

method rooted in topological data analysis, that makes neural networks more fathomable. Our method defines a measure of neural network complexity that correlates with deep learning best practices and the network's generalisation capabilities. The latter observation permits us to develop an early-stopping criterion that does not necessitate any validation data set.

Being one of the pioneering works on the investigation of neural networks using persistent homology, as illustrated in a recent survey by Hensel et al. [106], the field of topological machine learning experienced substantial growth over the last years. We are convinced that TDA will continue to play an important role in machine learning and outline several directions that we consider to be of interest. First, neural persistence may be used as a "self-regularisation" term to direct the updates of neural network parameters in such a way that `NP` is increased. Given the observed correlation between `NP` and generalisability, we hypothesise that such a regularisation term may lead to faster convergence. Similarly, increased `NP` may be used as an objective in neural architecture search [76] to find new neural network architectures. A prerequisite for this, however, is the extension of our proposed method to more sophisticated architectures such as attention layers. Such an enhancement requires a principled mapping of large-scale neural network architectures to a topology-based framework, a highly non-trivial endeavour. One way of addressing this is by adopting a "weight space" perspective, i.e. a perspective in which the complete network is reduced to its weights. This view is limited by the fact that the inductive biases from different architectures cannot be properly represented in the weight space alone. Great potential lies therefore in the search for more fruitful representations that balance expressivity and efficiency. A research direction that may serve as a link between real- and object-valued time series is to leverage a graph-based time series representation such as the visibility graph [140] for subsequent analysis. Visibility graphs capture important time series characteristics such as periodicity, and TDA may prove itself a useful analysis tool for a graph-based view on real-valued time series. Such a view may augment other recent works that use TDA for the analysis of real-valued time series [71, 293].

Outlook

There is no denying the fact that the ease with which data in the life sciences can be acquired will eventually lead to an accumulation of data sets of high temporal resolution. We hypothesise that almost every biomedical entity that can be measured repeatedly will be analysed in terms of its development over time. We will now briefly highlight some biomedical applications that may benefit from the development and deployment of novel time series analysis methods.

Single-Cell Analysis Over the last years, single-cell transcriptome sequencing (scRNA-seq) revolutionised modern biology by providing a new way of studying dynamic cellular processes [43]. Methods utilising scRNA-seq data enable researchers to discover new cell subpopulations [230] or help elucidate cell differentiation and maturation processes [192]. Trajectory inference (TI) methods are the core of many single-cell analysis approaches [219] that assign to each cell a so-called pseudotime, a numeric proxy for the cell's position within the dynamic process of interest. However, there is no guarantee that this proxy linearly correlates with true chronological time [268]. This discrepancy can be remedied by using synchronised cell populations in which each gene profile is annotated with a "real" time stamp. Since a cell can only be measured at *one* time point, it is crucial to construct "pseudocells" to prepare this data for time series analysis. We propose to use optimal transport (OT) (as introduced in Section 3.2.2) to match a cell's gene expression profile at time t with the closest profile at $t + 1$. As illustrated by Schiebinger et al. or Tong et al. [224, 262], one advantage of using OT is that it can act on the high-dimensional expression space directly. This alignment results in a multidimensional time series of gene expression profiles amenable to subsequence analyses such as `S3M` for the identification of gene expression motifs or extensions of `WTK` for time series-based cell clustering and the like. We may also view the resulting time series as an example of an object-valued time series (see Section 1) in which each cell consists of multiple gene expression "objects". This allows for the application of a principled analysis framework as laid out at the end of this chapter.

Organoids Another exciting field at the intersection of biology and medicine that may benefit from the utilisation of time series analysis tools is the study of organoids. Organoids are structures that exhibit functionalities and cellular architectures akin to in vivo organs. They develop from organ-specific progenitors or stem cells by means of a self-organisation process and serve as models that help improve our understanding of organ dysfunction and recovery [212]. Successfully developed organoids include models of the colon, liver, pancreas, or kidney, to name just a few (Rossi et al. [212] provide a thorough overview of the successes and the potential of the field). What makes organoids relevant for the field of biomedicine is that they can be derived from human stem cells and can simulate human pathologies at the organ level. Methods such as the ones developed in this thesis will play an important role in investigating an organoid's development process. In particular, subsequence-based analysis methods such as `S3M` are a promising way to detect novel gene expression motifs (i.e. temporal dependencies of up- and downregulation) associated with diseased or healthy organoids. Moreover, combining genetic trajectories with structural and morphological changes could generate insights into more fundamental biological questions concerning organ formation.

Lastly, the fact that organoids are able to model human pathologies makes them a vehicle to be used for drug screening studies. By tracking the temporal dynamics of a genome, methods such as `S3M` would allow researchers to determine when (i.e. at which genetical configuration) a drug might be most effective.

Complex Phenotypes Throughout this thesis, we analysed complex pathological conditions such as sepsis or myocardial ischaemia. In the case of sepsis, its complexity arises from the involvement of multiple organ systems that determine the presence of the condition, exacerbating a satisfactory definition of sepsis itself [178]. Similarly, diagnosing exercise-induced myocardial ischaemia requires performing a complex diagnostic procedure leading to exceedingly long signals in which relevant patterns may develop transiently. While from a medical perspective fairly different, the similarity of both conditions lies in the fact that typically, exceptionally long time series are available for their detection. High sampling rates result in long stress test ECGs; long ICU stays result in continuous monitoring data that may cover months. Future time series classification algorithms should therefore be efficient feature extractors that are able to find short signals and learn long-term dependencies from sequences consisting of hundreds of thousands of measurements. In addition, it may become critical to design algorithms that make no assumption about the length of the input signal at all (i.e. infinite horizon approaches [242]). Especially with increased usage of real-time monitoring, challenges arising from infinitely long data streams (e.g. distribution changes and the need for continuous parameter updates) will become more prevalent. While a steady pattern matching approach using shapelets could be a prudent first step, it seems more promising to learn patient representations that can perpetually be updated. In the case of patient representations derived from ECGs, we may learn representations from a small number of heartbeats and clinical variables. For sepsis, a patient's health status may be summarised by a function of the most relevant vital parameters, medication, and static clinical data. In both cases, we take a more patient-centric standpoint in which individual time series measurements (class I and II) become a mere building block of a more integrated and comprehensive representation (class III).

From this standpoint, it becomes crucial to shift from a perspective evolving around real-valued time series to a focus on object-valued time series. Notice that the latter view encompasses and generalises the previous one. This encourages researchers to develop a holistic view of the investigated objects, in which an object's temporal trajectory summarises the change of its individual characteristics in a *representative* manner. The meaning of "representative" in this context will be highly domain-dependent. Neural persistence exemplifies

an analysis tool for object-valued time series whose development followed a methodological framework of general applicability to the life sciences and biomedicine. We believe that such a framework may be of use as a guideline for the principled development of future analysis tools. Based on the notation in Definition 1.3, this framework should consist of the following steps:

1. Define a set of objects (e.g. O and $\neg O$) that are sufficiently different from each other.
2. Identify a metric space $(\mathcal{Z}, d)$, with distance function $d\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, that may recover these differences.
3. Use a well-defined map $z\colon \mathcal{Q} \to \mathcal{Z}$ to map all objects from $\mathcal{Q}$ to $\mathcal{Z}$.
4. Ensure that $d(z(O), z(\neg O))$ is sufficiently large.
5. Use $\{d(z(O_1), z(O_2)), \ldots, d(z(O_{m-1}), z(O_m))\}$ as representative time series for further analysis.

While these steps are reminiscent of general dimensionality reduction approaches, there is no *need* to reduce an object's dimensionality if $\mathcal{Z}$ is endowed with an appropriate distance function. That being said, one of the challenges in the implementation of the proposed framework is the identification of an appropriate metric space along with a suitable function for embedding objects into said space. The flexibility of artificial neural networks allows us to define the dimensionality of $\mathcal{Z}$ (often called the embedding space) and learn both the map z and a distance function d simultaneously. In particular, the hidden representations of autoencoders provide a well-defined mapping into $\mathcal{Z}$ that can be learnt alongside the distance function. When learning the map z, it should

a) satisfy the property of sequential continuity (i.e. $\lim_{\tau \to 0} z(O_{x+\tau}) = z(O_x)$ if $\lim_{\tau \to 0} O_{x+\tau} = O_x$), and

b) reflect the *sparse* labels acquired in Step 1.

The first property is important to ensure learning an embedding that will lead to a smooth trajectory. One way to "nudge" the network to learn such trajectories is to impose a certain structure on $\mathcal{Z}$. For example, we could restrict z to map its input on a sphere or hyperbolic geometries like the Poincaré disk. The label sparsity is due to the assumption that the differences in the objects from Step 1 are hard to determine and require deep domain knowledge. Depending on the level of sparsity, an active learning approach [286] may be incorporated. When there is no apparent choice for a distance function, which is the case when we rely on

a neural network to learn z, a metric learning approach [249] will guide the network to *learn* a useful distance function. This way, it is possible to learn Steps 1 through 4 in an end-to-end fashion with minimal user involvement. Lastly, if the task in Step 5 is well-defined and learnable by a neural network, it can also be incorporated into the pipeline.

As a motivating example for the utility of this framework, we want to briefly discuss a biomedical application where an object-valued view might be beneficial. This is the case in the analysis of electronic health records, where we may be interested in learning a patient's health status on any given day. What we consider healthy will depend on our field of expertise or interest and lead to labels that are best acquired in an active learning setting. An appropriate representation of the vast amounts of the per-day information in combination with an expressive distance function will allow us to 1) assess an individual patient's trajectory, and 2) compare the health state of different patients. The latter can be achieved in a per-day fashion or by comparing trajectories as a whole. More generally, this framework will make all kinds of dynamic processes amenable to an improved description that by construction aligns with the research question at hand.

To conclude, the increased availability of biological and biomedical time series data will change how we characterise and analyse specimens and research subjects as a whole. Taking their temporal development into account will allow scholars to develop a more holistic view of their field that may lead to novel biomedical insights. Machine learning methods — specifically, interpretable and explainable ones — can help facilitate this switch in perspective by providing powerful machinery to help identify the hidden motifs and manifolds that constitute life.

Acronyms

`NP`	Neural Persistence
`S3M`	Statistically Significant Subsequence Mining
`NIP`	Neural Ischaemia Prediction
`WTK`	Wasserstein Subsequence Kernel
ACS	Acute Coronary Syndrome
ANN	Artificial Neural Network
AUPRC	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic
CAD	Coronary Artery Disease
CD	Critical Difference
CDF	Cumulative Density Function
CI	Confidence Interval
CNN	Convolutional Neural Network
CVD	Cardiovascular Death
DTW	Dynamic Time Warping
ECG	Electrocardiogram
EHR	Electronic Health Record
EIMI	Exercise-Induced Myocardial Ischaemia
fMRI	Functional Magnetic Resonance Imaging
FNR	False Negative Rate
FPR	False Positive Rate
FWER	Family-wise Error Rate
gRSF	Generalized Random Shapelet Forests
HRV	Heart Rate Variability
ICU	Intensive Care Unit
MIMIC	Medical Information Mart for Intensive Care
ML	Machine Learning
MPS	Myocardial Perfusion Scan
MTL	Multi-Task Learning

OR	Odds Ratio
OT	Optimal Transport
PD	Persistence Diagram
PH	Persistent Homology
PR	Precision-Recall
PSD	Positive Semi-Definite
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RKHS	Reproducing Kernel Hilbert Space
RKKS	Reproducing Kernel Kreĭn Space
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RR	Respiratory Rate
SHAP	SHapley Additive exPlanations
SI	Suspected Infection
SIRS	Systemic Inflammatory Response Syndrome
SOFA	Sequential Organ Failure Assessment
SOTA	State-of-the-art
SPECT	Single-Photon Emission Computerized Tomography
SPM	Significant Pattern Mining
SVM	Support Vector Machine
TDA	Topological Data Analysis
TSC	Time Series Classification
YLL	Year of Life Lost

List of Figures

List of Tables

Bibliography

1. A. Achille and S. Soatto. "Emergence of invariance and disentanglement in deep representations". *Journal of Machine Learning Research* 18, 2018, pp. 1–34.
2. P. S. Addison. "Wavelet transforms and the ECG: a review". 26:5, 2005, R155–R199. DOI: 10.1088/0967-3334/26/5/r01.
3. S. R. Aghabozorgi, A. S. Shirkhorshidi, and Y. W. Teh. "Time-series clustering - A decade review". *Inf. Syst.* 53, 2015, pp. 16–38. DOI: 10.1016/j.is.2015.04.007.
4. S. Ahmad, T. Ramsay, L. Huebsch, S. Flanagan, S. McDiarmid, I. Batkin, L. McIntyre, S. R. Sundaresan, D. E. Maziak, F. M. Shamji, P. Hebert, D. Fergusson, A. Tinmouth, and A. J. E. Seely. "Continuous Multi-Parameter Heart Rate Variability Analysis Heralds Onset of Sepsis in Adults". *PLOS ONE* 4:8, 2009, pp. 1–10. DOI: 10.1371/journal.pone.0006642.
5. S. Alaee, K. Kamgar, and E. J. Keogh. "Matrix Profile XXII: Exact Discovery of Time Series Motifs under DTW". In: *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*. Ed. by C. Plant, H. Wang, A. Cuzzocrea, C. Zaniolo, and X. Wu. IEEE, 2020, pp. 900–905. DOI: 10.1109/ICDM50108.2020.00099.
6. J. Altschuler, J. Weed, and P. Rigollet. "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 30. Curran Associates, Inc., 2017, pp. 1964–1974.
7. S. Aminikhanghahi and D. J. Cook. "A survey of methods for time series change point detection". *Knowl. Inf. Syst.* 51:2, 2017, pp. 339–367. DOI: 10.1007/s10115-016-0987-z.
8. E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er. "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia". *Nature biotechnology* 31:6, 2013, pp. 545–552.

9. S. Ansari, N. Farzaneh, M. Duda, K. Horan, H. B. Andersson, Z. D. Goldberger, B. K. Nallamothu, and K. Najarian. "A Review of Automated Methods for Detection of Myocardial Ischemia and Infarction Using Electrocardiogram and Electronic Health Records". *IEEE reviews in biomedical engineering* 10, 2017, pp. 264–298.
10. N. Aronszajn. "Theory of reproducing kernels". *Transactions of the American mathematical society* 68:3, 1950, pp. 337–404.
11. A. Bagnall, J. Lines, J. Hills, and A. Bostrom. "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles". *IEEE Transactions on Knowledge and Data Engineering* 27:9, 2015, pp. 2522–2535.
12. A. J. Bagnall, J. Lines, A. Bostrom, J. Large, and E. J. Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". *Data Min. Knowl. Discov.* 31:3, 2017, pp. 606–660. DOI: 10.1007/s10618-016-0483-9.
13. D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations (ICLR)*. 2015.
14. R. B. Bapat and T. E. S. Raghavan. *Nonnegative matrices and applications*. Cambridge University Press, Cambridge, UK, 1997.
15. Z. Bar-Joseph, A. Gitter, and I. Simon. "Studying and modelling dynamic biological processes using time-series gene expression data". *Nature Reviews Genetics* 13:8, 2012, pp. 552–564.
16. G. E. Batista, X. Wang, and E. J. Keogh. "A complexity-invariant distance measure for time series". In: *SDM*. 2011, pp. 699–710.
17. J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. "Iterative Bregman Projections for Regularized Transportation Problems". *SIAM Journal on Scientific Computing* 37:2, 2015, A1111–A1138.
18. J. R. Bence. "Analysis of short time series: correcting for autocorrelation". *Ecology* 76:2, 1995, pp. 628–639.
19. Y. Bengio. "Practical recommendations for gradient-based training of deep architectures". In: *Neural Networks: Tricks of the Trade*. Ed. by G. Montavon, G. B. Orr, and K.-R. Müller. Vol. 7700. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, 2012, pp. 437–478.

20. D. J. Berndt and J. Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, USA, July 1994. Technical Report WS-94-03*. Ed. by U. M. Fayyad and R. Uthurusamy. AAAI Press, 1994, pp. 359–370.

21. M. Bianchini and F. Scarselli. "On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures". *IEEE Transactions on Neural Networks and Learning Systems* 25:8, 2014, pp. 1553–1565.

22. O. Biran and C. Cotton. "Explanation and justification in machine learning: A survey". In: *IJCAI-17 workshop on explainable AI (XAI)*. Vol. 8. 1. 2017, pp. 8–13.

23. D. W. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag. "What is the State of Neural Network Pruning?" In: *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. Ed. by I. S. Dhillon, D. S. Papailiopoulos, and V. Sze. mlsys.org, 2020. URL: https://proceedings.mlsys.org/book/296.pdf.

24. A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. "A Review on Outlier/Anomaly Detection in Time Series Data". *ACM Comput. Surv.* 54:3, 2021, 56:1–56:33. DOI: 10.1145/3444690.

25. A. S. Blevins and D. S. Bassett. "Topology in Biology". In: *Handbook of the Mathematics of the Arts and Sciences*. Ed. by B. Sriraman. Springer, Cham, Switzerland, 2020, pp. 1–23. DOI: 10.1007/978-3-319-70658-0_87-1.

26. C. Bock, T. Gumbsch, M. Moor, B. Rieck, D. Roqueiro, and K. Borgwardt. "Association mapping in biomedical time series via statistically significant shapelet mining". *Bioinformatics* 34:13, 2018, pp. i438–i446. DOI: 10.1093/bioinformatics/bty246.

27. C. Bock, M. Moor, C. R. Jutzeler, and K. Borgwardt. "Machine learning for biomedical time series classification: from shapelets to deep learning". In: *Artificial Neural Networks*. Springer, 2021, pp. 33–71. DOI: 10.1007/978-1-0716-0826-5_2.

28. C. Bock, B. Rieck, J. Walter, I. Strebel, K. Borgwardt, and C. Müller. "Cardiologist-level prediction of stress-induced myocardial ischemia using multi-task learning." In preparation.

29. C. Bock, M. Togninalli, E. Ghisu, T. Gumbsch, B. Rieck, and K. Borgwardt. "A Wasserstein Subsequence Kernel for Time Series". In: *2019 IEEE International Conference on Data Mining (ICDM)*. 2019, pp. 964–969. DOI: 10.1109/ICDM.2019.00108.

30. R. C. Bone, R. A. Balk, F. B. Cerra, R. P. Dellinger, A. M. Fein, W. A. Knaus, R. M. Schein, and W. J. Sibbald. "Definitions for Sepsis and Organ Failure and Guidelines for the Use of Innovative Therapies in Sepsis". *Chest* 101:6, 1992, pp. 1644–1655. ISSN: 0012-3692. DOI: https://doi.org/10.1378/chest.101.6.1644.

31. C. E. Bonferroni. "Teoria statistica delle classi e calcolo delle probabilità". *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commerciali di Firenze* 8, 1936, pp. 3–62.

32. K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. "Integrating structured biological data by Kernel Maximum Mean Discrepancy". *Bioinformatics* 22:14, 2006, e49–e57. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btl242.

33. K. M. Borgwardt. "Graph kernels". PhD thesis. Ludwig Maximilians University Munich, Germany, 2007. URL: http://edoc.ub.uni-muenchen.de/archive/00007169/.

34. B. E. Boser, I. Guyon, and V. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*. Ed. by D. Haussler. ACM, 1992, pp. 144–152. DOI: 10.1145/130385.130401.

35. A. Bostrom and A. J. Bagnall. "Binary Shapelet Transform for Multiclass Time Series Classification". In: *Big Data Analytics and Knowledge Discovery - 17th International Conference, DaWaK 2015, Valencia, Spain, September 1-4, 2015, Proceedings*. Ed. by S. Madria and T. Hara. Vol. 9263. Lecture Notes in Computer Science. Springer, 2015, pp. 257–269. DOI: 10.1007/978-3-319-22729-0_20.

36. G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

37. G. E. Bredon. *Topology and geometry*. Vol. 139. Springer Science & Business Media, 2013.

38. L. Breiman. "Random Forests". *Mach. Learn.* 45:1, 2001, pp. 5–32. DOI: 10.1023/A:1010933404324.

39. E. N. Brown, R. E. Kass, and P. P. Mitra. "Multiple neural spike train data analysis: state-of-the-art and future challenges". *Nature neuroscience* 7:5, 2004, pp. 456–461.

40. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

41. P. Bubenik. "Statistical topological data analysis using persistence landscapes". *Journal of Machine Learning Research* 16, 2015, pp. 77–102.

42. Y.-H. Byeon, S.-B. Pan, and K.-C. Kwak. "Intelligent deep models based on scalograms of electrocardiogram signals for biometrics". *Sensors* 19:4, 2019, p. 935.

43. R. Cannoodt, W. Saelens, and Y. Saeys. "Computational methods for trajectory inference from single-cell transcriptomics". *European journal of immunology* 46:11, 2016, pp. 2496–2506.

44. G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. "On the Local Behavior of Spaces of Natural Images". *International Journal of Computer Vision* 76:1, 2008, pp. 1–12.

45. C. J. Carstens and K. J. Horadam. "Persistent homology of collaboration networks". *Mathematical Problems in Engineering* 2013, 815035, 2013, pp. 1–7.

46. R. Caruana, S. Baluja, T. Mitchell, et al. "Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation". *Advances in neural information processing systems*, 1996, pp. 959–965.

47. R. Caruana. "Multitask Learning: A Knowledge-Based Source of Inductive Bias". In: *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, 1993, pp. 41–48.

48. F. Casacuberta, E. Vidal, and H. Rulot. "On the metric properties of dynamic time warping". *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35:11, 1987, pp. 1631–1633.

49. F. M. de Castilho, A. L. P. Ribeiro, V. Nobre, G. Barros, and M. R. de Sousa. "Heart rate variability as predictor of mortality in sepsis: A systematic review". *PLOS ONE* 13:9, 2018, pp. 1–13. DOI: 10.1371/journal.pone.0203487.

50. T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, W. Xie, G. L. Rosen, B. J. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. E. Carpenter, A. Shrikumar, J. Xu, E. M. Cofer, C. A. Lavender, S. C. Turaga, A. M. Alexandri, Z. Lu, D. J. Harris, D. DeCaprio, Y. Qi, A. Kundaje, Y. Peng, L. K. Wiley, M. H. Segler, S. M. Boca, S. J. Swamidass, A. Huang, A. Gitter, and C. S. Greene. "Opportunities and obstacles for deep learning in biology and medicine". *Journal of The Royal Society Interface* 15:141, 2018, p. 20170387.

51. F. Chollet et al. *Keras*. https://keras.io. 2015.

52. S. Chowdhury, T. Gebhart, S. Huntsman, and M. Yutin. "Path Homologies of Deep Feedforward Networks". In: *18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, December 16-19, 2019*. Ed. by M. A. Wani, T. M. Khoshgoftaar, D. Wang, H. Wang, and N. Seliya. IEEE, 2019, pp. 1077–1082. DOI: 10.1109/ICMLA.2019.00181.

53. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. "Extending persistence using Poincaré and Lefschetz duality". *Foundations of Computational Mathematics* 9:1, 2009, pp. 79–103.

54. D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. "Lipschitz functions have L_p-stable persistence". *Foundations of Computational Mathematics* 10:2, 2010, pp. 127–139.

55. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. 3rd. MIT Press, Cambridge, MA, USA, 2009.

56. C. A. Corneanu, S. Escalera, and A. M. Martínez. "Computing the Testing Error Without a Testing Set". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 2674–2682. DOI: 10.1109/CVPR42600.2020.00275.

57. J.-P. Couderc. "The telemetric and Holter ECG warehouse initiative (THEW): a data repository for the design, implementation and validation of ECG-related technologies". In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE. 2010, pp. 6252–6255.

58. M. Cuturi. "Fast global alignment kernels". In: *28th International Conference on Machine Learning (ICML)*. 2011, pp. 929–936.

59. M. Cuturi. "Permanents, transportation polytopes and positive definite kernels on histograms". In: *IJCAI*. 2007, pp. 732–737.

60. M. Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 26. Curran Associates, Inc., 2013, pp. 2292–2300.

61. M. Cuturi and M. Blondel. *Soft-DTW: a Differentiable Loss Function for Time-Series*. 2018. arXiv: 1703.01541 [stat.ML].

62. M. Cuturi and A. Doucet. "Autoregressive kernels for time series". *arXiv e-prints*, arXiv:1101.0673, 2011. arXiv: 1101.0673 [stat.ML].

63. M. Cuturi, J.-P. Vert, Ø. Birkenes, and T. Matsui. "A kernel for time series based on global alignments". In: *ICASSP*. Vol. 2. 2007, pp. 413–416.

64. J. M. Dale, L. Popescu, and P. D. Karp. "Machine learning methods for metabolic pathway prediction". *BMC Bioinform.* 11, 2010, p. 15. DOI: 10.1186/1471-2105-11-15.

65. M. R. Daliri. "Kernel earth mover's distance for EEG classification". *Clinical EEG and Neuroscience* 44:3, 2013, pp. 182–187.

66. H. A. Dau, A. J. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. J. Keogh. "The UCR time series archive". *IEEE CAA J. Autom. Sinica* 6:6, 2019, pp. 1293–1305. DOI: 10.1109/jas.2019.1911747.

67. H. A. Dau and E. J. Keogh. "Matrix Profile V: A Generic Technique to Incorporate Domain Knowledge into Motif Discovery". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017, pp. 125–134. DOI: 10.1145/3097983.3097993.

68. J. G. De Gooijer and R. J. Hyndman. "25 years of time series forecasting". *International journal of forecasting* 22:3, 2006, pp. 443–473.

69. J. Demšar. "Statistical comparisons of classifiers over multiple data sets". *Journal of Machine Learning Research* 7, 2006, pp. 1–30.

70. T. Desautels, J. Calvert, J. Hoffman, M. Jay, Y. Kerem, L. Shieh, D. Shimabukuro, U. Chettipally, M. D. Feldman, C. Barton, D. J. Wales, and R. Das. "Prediction of Sepsis in the Intensive Care Unit With Minimal Electronic Health Record Data: A Machine Learning Approach". *JMIR Med Inform* 4:3, 2016, e28. DOI: 10.2196/medinform.5909.

71. M. Dindin, Y. Umeda, and F. Chazal. "Topological Data Analysis for Arrhythmia Detection Through Modular Neural Networks". In: *Advances in Artificial Intelligence - 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13-15, 2020, Proceedings*. Ed. by C. Goutte and X. Zhu. Vol. 12109. Lec-

ture Notes in Computer Science. Springer, 2020, pp. 177–188. DOI: 10.1007/978-3-030-47358-7_17.

72. J. Durbin and G. S. Watson. "Testing for serial correlation in least squares regression: I". *Biometrika* 37:3/4, 1950, pp. 409–428.

73. D. Duvenaud, D. Maclaurin, and R. Adams. "Early Stopping as Nonparametric Variational Inference". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by A. Gretton and C. C. Robert. Vol. 51. Proceedings of Machine Learning Research. PMLR, Cadiz, Spain, 2016, pp. 1070–1077. URL: http://proceedings.mlr.press/v51/duvenaud16.html.

74. H. Edelsbrunner and J. Harer. *Computational topology: An introduction*. American Mathematical Society, Providence, RI, USA, 2010.

75. H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. "Topological persistence and simplification". *Discrete & Computational Geometry* 28:4, 2002, pp. 511–533.

76. T. Elsken, J. H. Metzen, and F. Hutter. "Neural Architecture Search: A Survey". *Journal of Machine Learning Research* 20:55, 2019, pp. 1–21. URL: http://jmlr.org/papers/v20/18-598.html.

77. W. Enders. *Applied econometric time series*. John Wiley & Sons, 2008.

78. C. Faloutsos, J. Gasthaus, T. Januschowski, and Y. Wang. "Forecasting Big Time Series: Old and New". *Proc. VLDB Endow.* 11:12, 2018, pp. 2102–2105. DOI: 10.14778/3229863.3229878.

79. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. "Deep learning for time series classification: a review". *Data Mining and Knowledge Discovery*, 2019, pp. 1–47.

80. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller. "Deep learning for time series classification: a review". *Data Min. Knowl. Discov.* 33:4, 2019, pp. 917–963. DOI: 10.1007/s10618-019-00619-1.

81. T. Fawcett. "An introduction to ROC analysis". *Pattern Recognit. Lett.* 27:8, 2006, pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

82. A. Feragen, F. Lauze, and S. Hauberg. "Geodesic exponential kernels: When curvature and linearity conflict". In: *IEEE CVPR*. 2015, pp. 3032–3042.

83. G. Fiscon, E. Weitschek, A. Cialini, G. Felici, P. Bertolazzi, S. De Salvo, A. Bramanti, P. Bramanti, and M. C. De Cola. "Combining EEG signal processing with supervised methods for Alzheimer's patients classification". *BMC medical informatics and decision making* 18:1, 2018, pp. 1–10.

84. R. A. Fisher. "Statistical methods for research workers". In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

85. R. Flamary and N. Courty. *POT Python Optimal Transport library*. 2017. URL: https://github.com/rflamary/POT.

86. J. B. J. baron Fourier. *Théorie analytique de la chaleur*. Chez Firmin Didot, père et fils, 1822.

87. J. Frankle and M. Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=rJl-b3RcF7.

88. T. Gebhart, P. Schrater, and A. Hylton. "Characterizing the Shape of Activation Space in Deep Neural Networks". In: *18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, December 16-19, 2019*. Ed. by M. A. Wani, T. M. Khoshgoftaar, D. Wang, H. Wang, and N. Seliya. IEEE, 2019, pp. 1537–1542. DOI: 10.1109/ICMLA.2019.00254.

89. M. F. Ghalwash and Z. Obradovic. "Early classification of multivariate temporal observations by extraction of interpretable shapelets". *BMC Bioinform.* 13, 2012, p. 195. DOI: 10.1186/1471-2105-13-195.

90. S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, and E. Keogh. "Matrix Profile XII: MPdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios". In: *IEEE International Conference on Data Mining (ICDM)*. 2018, pp. 965–970.

91. S. Gharghabi, S. Imani, A. J. Bagnall, A. Darvishzadeh, and E. J. Keogh. "Matrix Profile XII: MPdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios". In: *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 965–970. DOI: 10.1109/ICDM.2018.00119.

92. A. Goldberger. *Goldberger's Clinical Electrocardiography*. Elsevier, 2018.

93. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

94. D. Gordon, D. Hendler, A. Kontorovich, and L. Rokach. "Local-shapelets for fast classification of spectrographic measurements". *Expert Syst. Appl.* 42:6, 2015, pp. 3150–3158. DOI: 10.1016/j.eswa.2014.11.043.

95. J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. "Learning time-series shapelets". In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. Ed. by S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani. ACM, 2014, pp. 392–401. DOI: 10.1145/2623330.2623613.

96. A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. "A Kernel Two-Sample Test". *J. Mach. Learn. Res.* 13, 2012, pp. 723–773. URL: http://dl.acm.org/citation.cfm?id=2188410.

97. T. Gumbsch, C. Bock, M. Moor, B. Rieck, and K. Borgwardt. "Enhancing statistical power in temporal biomarker discovery through representative shapelet mining". *Bioinformatics* 36:Supplement_2, 2020, pp. i840–i848. DOI: 10.1093/bioinformatics/btaa815.

98. A. Gumpinger. "Machine learning on molecular networks to decipher the genetics underlying complex traits". PhD thesis. ETH Zurich, 2020.

99. W. H. Guss and R. Salakhutdinov. "On Characterizing the Capacity of Neural Networks using Algebraic Topology". *arXiv preprint arXiv:1802.04443*, 2018.

100. B. Haasdonk. "Feature space interpretation of SVMs with indefinite kernels". *IEEE TPAMI* 27:4, 2005, pp. 482–492.

101. A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng. "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network". *Nature medicine* 25:1, 2019, p. 65.

102. K. Hara, D. Saitoh, and H. Shouno. "Analysis of dropout learning regarded as ensemble learning". In: *Artificial Neural Networks and Machine Learning (ICANN)*. Ed. by A. E. Villa, P. Masulli, and A. J. Pons Rivero. Lecture Notes in Computer Science 9887. Springer, Cham, Switzerland, 2016, pp. 72–79.

103. H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan. "Multitask learning and benchmarking with clinical time series data". *Scientific data* 6:1, 2019, pp. 1–18.

104. D. Haussler. *Convolution kernels on discrete structures*. Technical report. Technical report, Department of Computer Science, University of California, Santa Cruz, 1999.

105. K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

106. F. Hensel, M. Moor, and B. Rieck. "A Survey of Topological Machine Learning Methods". *Frontiers Artif. Intell.* 4, 2021, p. 681108. DOI: 10.3389/frai.2021.681108.

107. F. L. Hitchcock. "The distribution of a product from several sources to numerous localities". *Journal of mathematics and physics* 20:1-4, 1941, pp. 224–230.

108. S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". *Neural Comput.* 9:8, 1997, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

109. C. D. Hofer, F. Graf, B. Rieck, M. Niethammer, and R. Kwitt. "Graph Filtration Learning". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4314–4323. URL: http://proceedings.mlr.press/v119/hofer20b.html.

110. C. D. Hofer, R. Kwitt, and M. Niethammer. "Learning Representations of Persistence Barcodes". *J. Mach. Learn. Res.* 20, 2019, 126:1–126:45. URL: http://jmlr.org/papers/v20/18-358.html.

111. C. D. Hofer, R. Kwitt, M. Niethammer, and M. Dixit. "Connectivity-Optimized Representation Learning via Persistent Homology". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2751–2760. URL: http://proceedings.mlr.press/v97/hofer19a.html.

112. C. D. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. "Deep learning with topological signatures". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 1633–1643.

113. D. Horak, S. Maletić, and M. Rajković. "Persistent homology of complex networks". *Journal of Statistical Mechanics: Theory and Experiment* 2009:03, 2009, P03034.

114. M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt. "Set Functions for Time Series". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4353–4363. arXiv: 1909.12064 [cs.LG].

115. R. S. Hotchkiss, L. L. Moldawer, S. M. Opal, K. Reinhart, I. R. Turnbull, and J.-L. Vincent. "Sepsis and septic shock". *Nature reviews Disease primers* 2:1, 2016, pp. 1–21.

116. W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. "Open Graph Benchmark: Datasets for Machine Learning on Graphs". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

117. S. L. Hyland, M. Faltys, M. Hüser, X. Lyu, T. Gumbsch, C. Esteban, C. Bock, M. Horn, M. Moor, B. Rieck, et al. "Early prediction of circulatory failure in the intensive care unit using machine learning". *Nature medicine* 26:3, 2020, pp. 364–373. DOI: 10.1038/s41591-020-0789-4.

118. S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 448–456.

119. A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. "Perceiver: General Perception with Iterative Attention". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4651–4664. URL: http://proceedings.mlr.press/v139/jaegle21a.html.

120. S. L. James, D. Abate, K. H. Abate, S. M. Abay, C. Abbafati, N. Abbasi, H. Abbastabar, F. Abd-Allah, J. Abdela, A. Abdelalim, et al. "Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017". *The Lancet* 392:10159, 2018, pp. 1789–1858.

121. J. Ji, X. Chen, C. Luo, and P. Li. "A deep multi-task learning approach for ECG data analysis". In: *2018 IEEE EMBS International Conference on Biomedical & Health Informatics, BHI 2018, Las Vegas, NV, USA, March 4-7, 2018*. IEEE, 2018, pp. 124–127. DOI: 10.1109/BHI.2018.8333385.

122. A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark IV. "MIMIC-IV (version 1.0)". *PhysioNet*, 2020.

123. A. E. W. Johnson, D. J. Stone, L. A. Celi, and T. J. Pollard. "The MIMIC Code Repository: enabling reproducibility in critical care research". *Journal of the American Med-*

ical Informatics Association 25:1, 2017, pp. 32–39. ISSN: 1527-974X. DOI: 10.1093/jamia/ocx084.

124. A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. "MIMIC-III, a freely accessible critical care database". *Scientific data* 3:1, 2016, pp. 1–9.

125. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. "Highly accurate protein structure prediction with AlphaFold". *Nature*, 2021. (Accelerated article preview). DOI: 10.1038/s41586-021-03819-2.

126. K. Karkazis. "The misuses of "biological sex"." *Lancet (London, England)* 394:10212, 2019, pp. 1898–1899.

127. I. Karlsson, P. Papapetrou, and H. Boström. "Generalized random shapelet forests". *Data Min. Knowl. Discov.* 30:5, 2016, pp. 1053–1085. DOI: 10.1007/s10618-016-0473-y.

128. R. J. Kate. "Using dynamic time warping distances as features for improved time series classification". *Data Min. Knowl. Discov.* 30:2, 2016, pp. 283–312. DOI: 10.1007/s10618-015-0418-x.

129. M. Khalilia, S. Chakraborty, and M. Popescu. "Predicting disease risks from highly imbalanced data using random forest". *BMC medical informatics and decision making* 11:1, 2011, pp. 1–13.

130. V. Khrulkov and I. Oseledets. "Geometry Score: A Method For Comparing Generative Adversarial Networks". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2621–2629.

131. P. Kidger and T. Lyons. "Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU". In: *International Conference on Learning Representations*. https://github.com/patrick-kidger/signatory. 2021.

132. W. H. Kim, N. Adluru, M. K. Chung, S. Charchut, J. J. GadElkarim, L. L. Altshuler, T. Moody, A. R. Kumar, V. Singh, and A. D. Leow. "Multi-resolutional Brain Network Filtering and Analysis via Wavelets on Non-Euclidean Space". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013 - 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part III*. Ed. by K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab. Vol. 8151. Lecture Notes in Computer Science. Springer, 2013, pp. 643–651. DOI: 10.1007/978-3-642-40760-4_80.

133. D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

134. N. Kitaev, L. Kaiser, and A. Levskaya. "Reformer: The Efficient Transformer". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkgNKkHtvB.

135. M. Kolar, L. Song, A. Ahmed, and E. P. Xing. "Estimating time-varying networks". *The Annals of Applied Statistics*, 2010, pp. 94–123.

136. S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. K. Rohde. "Generalized Sliced Wasserstein Distances". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett. 2019, pp. 261–272. URL: https://proceedings.neurips.cc/paper/2019/hash/f0935e4cd5920aa6c7c996a5ee53a70f-Abstract.html.

137. S. Kolouri, Y. Zou, and G. K. Rohde. "Sliced Wasserstein Kernels for Probability Distributions". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 5258–5267. DOI: 10.1109/CVPR.2016.568.

138. A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images", 2009.

139. A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed.

by P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. 2012, pp. 1106–1114. URL: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

140. L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño. "From time series to complex networks: The visibility graph". *Proceedings of the National Academy of Sciences* 105:13, 2008, pp. 4972–4975. ISSN: 0027-8424. DOI: 10.1073/pnas.0709247105.

141. D. Lagun, M. Ageev, Q. Guo, and E. Agichtein. "Discovering common motifs in cursor movement data for improving web search". In: *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*. Ed. by B. Carterette, F. Diaz, C. Castillo, and D. Metzler. ACM, 2014, pp. 183–192. DOI: 10.1145/2556195.2556265.

142. T. Le, M. Yamada, K. Fukumizu, and M. Cuturi. "Tree-Sliced Variants of Wasserstein Distances". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett. 2019, pp. 12283–12294. URL: https://proceedings.neurips.cc/paper/2019/hash/2d36b5821f8affc6868b59dfc9af6c9f-Abstract.html.

143. Y. LeCun, Y. Bengio, et al. "Convolutional networks for images, speech, and time series". *The handbook of brain theory and neural networks* 3361:10, 1995, p. 1995.

144. Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Comput.* 1:4, 1989, pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

145. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86:11, 1998, pp. 2278–2324.

146. D. Lee, X. Jiang, and H. Yu. "Harmonized representation learning on dynamic EHR graphs". *Journal of biomedical informatics* 106, 2020, p. 103426.

147. G. Lee, R. Twerenbold, Y. Tanglay, T. Reichlin, U. Honegger, M. Wagener, C. Jaeger, M. R. Gimenez, T. Hochgruber, C. Puelacher, et al. "Clinical benefit of high-sensitivity cardiac troponin I in the detection of exercise-induced myocardial ischemia". *American heart journal* 173, 2016, pp. 8–17.

148. D. Levin, T. Lyons, and H. Ni. *Learning from the past, predicting the statistics for the future, learning an evolving system*. 2016. arXiv: 1309.0260 [q-fin.ST].

149. T. W. Liao. "Clustering of time series data - a survey". *Pattern Recognit.* 38:11, 2005, pp. 1857–1874. DOI: 10.1016/j.patcog.2005.01.025.

150. U. de Lichtenberg, L. J. Jensen, S. Brunak, and P. Bork. "Dynamic complex formation during the yeast cell cycle". *science* 307:5710, 2005, pp. 724–727.

151. B. Lim and S. Zohren. "Time-series forecasting with deep learning: a survey". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379:2194, 2021, p. 20200209. DOI: 10.1098/rsta.2020.0209.

152. H.-T. Lin and C.-J. Lin. *A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods*. Technical report. National Taiwan University, 2003.

153. J. Lin, E. J. Keogh, S. Lonardi, and B. Y. Chiu. "A symbolic representation of time series, with implications for streaming algorithms". In: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD 2003, San Diego, California, USA, June 13, 2003*. Ed. by M. J. Zaki and C. C. Aggarwal. ACM, 2003, pp. 2–11. DOI: 10.1145/882082.882086.

154. J. Lines and A. Bagnall. "Time series classification with ensembles of elastic distance measures". *Data Mining and Knowledge Discovery* 29:3, 2015, pp. 565–592.

155. J. Lines, S. Taylor, and A. J. Bagnall. "HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification". In: *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*. Ed. by F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu. IEEE Computer Society, 2016, pp. 1041–1046. DOI: 10.1109/ICDM.2016.0133.

156. Z. C. Lipton. "The Mythos of Model Interpretability". *ACM Queue* 16:3, 2018, p. 30. DOI: 10.1145/3236386.3241340.

157. F. Llinares-López and K. Borgwardt. "Machine Learning for Biomarker Discovery: Significant Pattern Mining". In: *Analyzing Network Data in Biology and Medicine: An Interdisciplinary Textbook for Biological, Medical and Computational Scientists*. Cambridge University Press, 2019, pp. 313–368. DOI: 10.1017/9781108377706.009.

158. G. Loosli, S. Canu, and C. S. Ong. "Learning SVM in Kreĭn spaces". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38:6, 2015, pp. 1204–1216.

159. F. Lopez-Jimenez, Z. Attia, A. M. Arruda-Olson, R. Carter, P. Chareonthaitawee, H. Jouni, S. Kapa, A. Lerman, C. Luong, J. R. Medina-Inojosa, P. A. Noseworthy, P. A. Pellikka, M. M. Redfield, V. L. Roger, G. S. Sandhu, C. Senecal, and P. A. Friedman. "Artificial Intelligence in Cardiology: Present and Future". *Mayo Clinic Proceedings* 95:5, 2020, pp. 1015–1039. ISSN: 0025-6196. DOI: https://doi.org/10.1016/j.mayocp.2020.01.038.

160. H. A. Lorentz, A. Einstein, and H. Minkowski. *Das Relativitätsprinzip*. Springer Fachmedien Wiesbaden GmbH, 1923, p. 55. DOI: 10.1007/978-3-663-19510-8.

161. A. Lorincz, L. Attila Jeni, Z. Szabo, J. F. Cohn, and T. Kanade. "Emotional Expression Classification Using Time-Series Kernels". In: *IEEE CVPR Workshops*. 2013, pp. 889–895.

162. P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. "Extracting insights from the shape of complex data using topology". *Scientific Reports* 3, 2013, pp. 1–8.

163. S. M. Lundberg and S.-I. Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

164. R. Luss and A. d'Aspremont. "Support vector machine classification with indefinite kernels". *Mathematical Programming Computation* 1:2, 2009, pp. 97–118.

165. D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003, p. 139. ISBN: 978-0-521-64298-9.

166. M. Mahsereci, L. Balles, C. Lassner, and P. Hennig. "Early Stopping without a Validation Set". *CoRR* abs/1703.09580, 2017. arXiv: 1703.09580. URL: http://arxiv.org/abs/1703.09580.

167. D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel, and S. H. A. Chen. "NeuroKit2: A Python toolbox for neurophysiological signal processing". *Behavior Research Methods*, 2021. ISSN: 1554-3528. DOI: 10.3758/s13428-020-01516-y.

168. O. Maron and T. Lozano-Pérez. "A Framework for Multiple-Instance Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Jordan, M. Kearns, and S. Solla. Vol. 10. MIT Press, 1998. URL: https://proceedings.neurips.cc/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf.

169. P.-F. Marteau and S. Gibet. "On Recursive Edit Distance Kernels With Application to Time Series Classification". *IEEE Transactions on Neural Networks and Learning Systems* 26:6, 2015, pp. 1121–1133.

170. B. Mathias, J. Mira, and S. D. Larson. "Pediatric sepsis". *Current opinion in pediatrics* 28:3, 2016, p. 380.

171. A. Mazurie, D. Bonchev, B. Schwikowski, and G. A. Buck. "Evolution of metabolic network organization". *BMC Systems Biology* 4:1, 2010, pp. 1–10.

172. A. Mezari and I. Maglogiannis. "Gesture Recognition Using Symbolic Aggregate Approximation and Dynamic Time Warping on Motion Data". In: *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*. PervasiveHealth '17. Association for Computing Machinery, Barcelona, Spain, 2017, pp. 342–347. ISBN: 9781450363631. DOI: 10.1145/3154862.3154927.

173. D. Minnen, C. L. I. Jr., I. A. Essa, and T. Starner. "Discovering Multivariate Motifs using Subsequence Density Estimation and Greedy Mixture Learning". In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*. AAAI Press, 2007, pp. 615–620. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-097.php.

174. D. M. Mirvis and A. L. Goldberger. "Electrocardiography". *Heart Disease. A Textbook of Cardiovascular Medicine, 6th ed. Philadelphia: WB Saunders*, 2001, pp. 82–128.

175. G. Montavon, W. Samek, and K.-R. Müller. "Methods for interpreting and understanding deep neural networks". *Digital Signal Processing* 73, 2017, pp. 1–15.

176. K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. v. d. Elzen, M. J. Hirn, R. R. Coifman, N. B. Ivanova, G. Wolf, and S. Krishnaswamy. "Visualizing structure and transitions in high-dimensional biological data". *Nature Biotechnology* 37:12, 2019, pp. 1482–1492.

177. M. Moor, M. Horn, B. Rieck, and K. M. Borgwardt. "Topological Autoencoders". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054. URL: http://proceedings.mlr.press/v119/moor20a.html.

178. M. Moor, B. Rieck, M. Horn, C. R. Jutzeler, and K. Borgwardt. "Early Prediction of Sepsis in the ICU Using Machine Learning: A Systematic Review". *Frontiers in medicine* 8, 2021, p. 348.

179. A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, and M. B. Westover. "Exact Discovery of Time Series Motifs". In: *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*. SIAM, 2009, pp. 473–484. DOI: 10.1137/1.9781611972795.41.

180. J. R. Munkres. *Elements of algebraic topology*. CRC Press, Boca Raton, FL, USA, 1996.

181. P. D. Myers, B. M. Scirica, and C. M. Stultz. "Machine learning improves risk stratification after acute coronary syndrome". *Scientific reports* 7:1, 2017, pp. 1–12.

182. A. H. A. W. G. on Myocardial Segmentation, R. for Cardiac Imaging: M. D. Cerqueira, N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan, et al. "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association". *Circulation* 105:4, 2002, pp. 539–542.

183. S. Nagai, D. Anzai, and J. Wang. "Motion artefact removals for wearable ECG using stationary wavelet transform". *Healthcare technology letters* 4:4, 2017, pp. 138–141.

184. G. Naisat. "Tropical Algebra and Algebraic Topology of Deep Neural Networks". PhD thesis. The University of Chicago, 2020.

185. H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo. "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges". *Expert Syst. Appl.* 105, 2018, pp. 233–261. DOI: 10.1016/j.eswa.2018.03.056.

186. F. Odone, A. Barla, and A. Verri. "Building kernels from binary strings for image matching". *IEEE Trans. Image Process.* 14:2, 2005, pp. 169–180. DOI: 10.1109/TIP.2004.840701.

187. D. Oglic and T. Gärtner. "Learning in Reproducing Kernel Kreĭn Spaces". In: *ICML*. 2018, pp. 3859–3867.

188. S. J. Orfanidis. *Introduction to signal processing*. Pearson Education, Inc, 2016.

189. I. Osawa, T. Goto, Y. Yamamoto, and Y. Tsugawa. "Machine-learning-based prediction models for high-need high-cost patients using nationwide clinical and claims data". *NPJ digital medicine* 3:1, 2020, pp. 1–9.

190. R. S. Pathak. *The wavelet transform*. Vol. 4. Springer Science & Business Media, 2009.

191. O. Patri, M. Wojnowicz, and M. Wolff. "Discovering Malware with Time Series Shapelets". In: *50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, Hawaii, USA, January 4-7, 2017*. Ed. by T. Bui. ScholarSpace / AIS Electronic Library (AISeL), 2017, pp. 1–10. URL: http://hdl.handle.net/10125/41898.

192. F. Paul, Y. Arkin, A. Giladi, D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, D. Winter, D. Lara-Astiaso, M. Gury, A. Weiner, et al. "Transcriptional heterogeneity and lineage commitment in myeloid progenitors". *Cell* 163:7, 2015, pp. 1663–1677.

193. K. Pearson. "On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can be Reasonably Supposed to have Arisen from Random Sampling". In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by S. Kotz and N. L. Johnson. Springer New York, New York, NY, 1992, pp. 11–28. ISBN: 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9_2.

194. D. Pena, G. C. Tiao, and R. S. Tsay. *A course in time series analysis*. Vol. 322. John Wiley & Sons, 2011.

195. T. Pollehn, W. Brady, A. Perron, and F. Morris. "The electrocardiographic differential diagnosis of ST segment depression". *Emergency medicine journal: EMJ* 19:2, 2002, p. 129.

196. C. Puelacher, M. Wagener, R. Abächerli, U. Honegger, N. Lhasam, N. Schaerli, G. Prêtre, I. Strebel, R. Twerenbold, J. Boeddinghaus, et al. "Diagnostic value of ST-segment deviations during cardiac exercise stress testing: Systematic comparison of different ECG leads and time-points". *International journal of cardiology* 238, 2017, pp. 166–172.

197. K. Qiu, X. Wang, T. Li, and Y. Gu. "Graph-based reconstruction of time-varying spatial signals". In: *2016 IEEE International Conference on Digital Signal Processing (DSP)*. 2016, pp. 355–359. DOI: 10.1109/ICDSP.2016.7868578.

198. M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. "On the expressive power of deep neural networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 2847–2854.

199. A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al. "Scalable and accurate deep learning with electronic health records". *npj Digital Medicine* 1:1, 18, 2018.

200. P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. 2017. arXiv: 1711.05225 [cs.CV].

201. T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. "Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping". *ACM Trans. Knowl. Discov. Data* 7:3, 2013, 10:1–10:31. URL: https://dl.acm.org/citation.cfm?id=2500489.

202. B. G. Reed and B. R. Carr. "The normal menstrual cycle and the control of ovulation". *Endotext*, 2015. URL: https://www.ncbi.nlm.nih.gov/books/NBK279054/.

203. A. H. Ribeiro, M. H. Ribeiro, G. M. Paixão, D. M. Oliveira, P. R. Gomes, J. A. Canazart, M. P. Ferreira, C. R. Andersson, P. W. Macfarlane, M. Wagner Jr, et al. "Automatic diagnosis of the 12-lead ECG using a deep neural network". *Nature communications* 11:1, 2020, pp. 1–9.

204. B. Rieck. "Persistent Homology in Multivariate Data Visualization". PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2017. DOI: 10.11588/heidok.00022914.

205. B. Rieck, C. Bock, and K. Borgwardt. "A Persistent Weisfeiler-Lehman Procedure for Graph Classification". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5448–5458. URL: https://proceedings.mlr.press/v97/rieck19a.html.

206. B. Rieck, U. Fugacci, J. Lukasczyk, and H. Leitte. "Clique Community Persistence: A Topological Visual Analysis Approach for Complex Networks". *IEEE Transactions on Visualization and Computer Graphics* 24:1, 2018, pp. 822–831.

207. B. Rieck and H. Leitte. "Exploring and comparing clusterings of multivariate data sets using persistent homology". *Computer Graphics Forum* 35:3, 2016, pp. 81–90.

208. B. Rieck, M. Togninalli, C. Bock, M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt. "Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology". In: *International Conference on Learning Representations (ICLR)*. 2019. DOI: 10.3929/ethz-b-000327207.

209. B. Rieck, T. Yates, C. Bock, K. Borgwardt, G. Wolf, N. Turk-Browne, and S. Krishnaswamy. "Uncovering the Topology of Time-Varying fMRI Data using Cubical Persistence". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6900–6912. arXiv: 2006.07882 [q-bio.NC].

210. A. R. P. Riera, C. Ferreira, C. Ferreira Filho, M. Ferreira, A. Meneghini, A. H. Uchida, E. Schapachnik, S. Dubner, and L. Zhang. "The enigmatic sixth wave of the electrocardiogram: the U wave". *Cardiology journal* 15:5, 2008, pp. 408–421.

211. J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso. "Rotation Forest: A New Classifier Ensemble Method". *IEEE Trans. Pattern Anal. Mach. Intell.* 28:10, 2006, pp. 1619–1630. DOI: 10.1109/TPAMI.2006.211.

212. G. Rossi, A. Manfrin, and M. P. Lutolf. "Progress and potential in organoid research". *Nature Reviews Genetics* 19:11, 2018, pp. 671–687.

213. S. Roy, D. Mincu, E. Loreaux, A. Mottram, I. Protsyuk, N. Harris, Y. Xue, J. Schrouff, H. Montgomery, A. Connell, N. Tomasev, A. Karthikesalingam, and M. Seneviratne. "Multitask prediction of organ dysfunction in the intensive care unit using sequential subnetwork routing". *Journal of the American Medical Informatics Association*, 2021. ocab101. ISSN: 1527-974X. DOI: 10.1093/jamia/ocab101.

214. Y. Rubner, C. Tomasi, and L. J. Guibas. "The Earth Mover's Distance as a Metric for Image Retrieval". *Int. J. Comput. Vis.* 40:2, 2000, pp. 99–121. DOI: 10.1023/A:1026543900054.

215. S. Ruder. "An overview of multi-task learning in deep neural networks". *arXiv preprint arXiv:1706.05098*, 2017.

216. A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. J. Bagnall. "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances". *Data Min. Knowl. Discov.* 35:2, 2021, pp. 401–449. DOI: 10.1007/s10618-020-00727-3.

217. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*. Technical report. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

218. S. Rüping. *SVM kernels for time series analysis*. Technical report 43. Technical University of Dortmund, 2001.

219. W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. "A comparison of single-cell trajectory inference methods". *Nature biotechnology* 37:5, 2019, pp. 547–554.

220. H. Sakoe. "Dynamic-programming approach to continuous speech recognition". In: *1971 Proc. the International Congress of Acoustics, Budapest*. 1971.

221. A. Savitzky and M. J. Golay. "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry* 36:8, 1964, pp. 1627–1639.

222. A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. "On the information bottleneck theory of deep learning". In: *International Conference on Learning Representations (ICLR)*. 2018.

223. P. Schäfer. "The BOSS is concerned with time series classification in the presence of noise". *Data Min. Knowl. Discov.* 29:6, 2015, pp. 1505–1530. DOI: 10.1007/s10618-014-0377-7.

224. G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, et al. "Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming". *Cell* 176:4, 2019, pp. 928–943.

225. D. E. Schlesinger and C. M. Stultz. "Deep learning for cardiovascular risk stratification". *Current Treatment Options in Cardiovascular Medicine* 22:8, 2020, pp. 1–14.

226. B. Schölkopf, A. J. Smola, and K. Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". *Neural Comput.* 10:5, 1998, pp. 1299–1319. DOI: 10.1162/089976698300017467.

227. B. Schölkopf and A. J. Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002. ISBN: 9780262194754. URL: https://www.worldcat.org/oclc/48970254.

228. T. J. Sejnowski. "The unreasonable effectiveness of deep learning in artificial intelligence". *Proceedings of the National Academy of Sciences* 117:48, 2020, pp. 30033–30038.

229. P. Senin and S. Malinchik. "SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model". In: *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*. Ed. by H. Xiong, G. Karypis, B. M. Thuraisingham, D. J. Cook, and X. Wu. IEEE Computer Society, 2013, pp. 1175–1180. DOI: 10.1109/ICDM.2013.52.

230. A. K. Shalek, R. Satija, J. Shuga, J. J. Trombetta, D. Gennert, D. Lu, P. Chen, R. S. Gertner, J. T. Gaublomme, N. Yosef, et al. "Single-cell RNA-seq reveals dynamic paracrine control of cellular variation". *Nature* 510:7505, 2014, pp. 363–369.

231. D. Shanmugam, D. W. Blalock, and J. V. Guttag. "Multiple Instance Learning for ECG Risk Stratification". In: *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2019, 9-10 August 2019, Ann Arbor, Michigan, USA*. Ed. by F. Doshi-Velez, J. Fackler, K. Jung, D. C. Kale, R. Ranganath, B. C. Wallace, and J. Wiens. Vol. 106. Proceedings of Machine Learning Research. PMLR, 2019, pp. 124–139. URL: http://proceedings.mlr.press/v106/shanmugam19a.html.

232. T. Sharir, K. Merzon, I. Kruchin, A. Bojko, E. Toledo, A. Asman, and P. Chouraqui. "Use of electrocardiographic depolarization abnormalities for detection of stress-induced ischemia as defined by myocardial perfusion imaging". *The American journal of cardiology* 109:5, 2012, pp. 642–650.

233. N. Shervashidze and K. M. Borgwardt. "Fast subtree kernels on graphs". In: *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. Ed. by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Curran Associates, Inc., 2009, pp. 1660–1668. URL: https://proceedings.neurips.cc/paper/2009/hash/0a49e3c3a03ebde64f85c0bacd8a08e2-Abstract.html.

234. N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. "Weisfeiler-Lehman Graph Kernels". *J. Mach. Learn. Res.* 12, 2011, pp. 2539–2561. URL: http://dl.acm.org/citation.cfm?id=2078187.

235. R. Shwartz-Ziv and N. Tishby. "Opening the black box of deep neural networks via information". *arXiv preprint arXiv:1703.00810*, 2017.

236. D. F. Silva and G. E. A. P. A. Batista. "Elastic Time Series Motifs and Discords". In: *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*. Ed. by M. A. Wani, M. M. Kantardzic, M. S. Mouchaweh, J. Gama, and E. Lughofer. IEEE, 2018, pp. 237–242. DOI: 10.1109/ICMLA.2018.00042.

237. M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus. "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". *JAMA* 315:8, 2016, pp. 801–810. ISSN: 0098-7484. DOI: 10.1001/jama.2016.0287.

238. A. Sizemore, C. Giusti, and D. S. Bassett. "Classification of weighted networks through mesoscale homological features". *Journal of Complex Networks* 5:2, 2017, pp. 245–273.

239. J. Skardinga, B. Gabrys, and K. Musial. "Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey". *IEEE Access*, 2021.

240. N. V. Smirnov. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples". *Bull. Math. Univ. Moscou* 2:2, 1939, pp. 3–14.

241. A. A. Sodemann, M. P. Ross, and B. J. Borghetti. "A Review of Anomaly Detection in Automated Surveillance". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42:6, 2012, pp. 1257–1272. DOI: 10.1109/TSMCC.2012.2215319.

242. A. Solin, J. Hensman, and R. E. Turner. "Infinite-Horizon Gaussian Processes". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/b865367fc4c0845c0682bd466e6ebf4c-Paper.pdf.

243. S. M. Sou, C. Puelacher, R. Twerenbold, M. Wagener, U. Honegger, T. Reichlin, N. Schaerli, G. Pretre, R. Abächerli, C. Jaeger, et al. "Direct comparison of cardiac troponin I and cardiac troponin T in the detection of exercise-induced myocardial ischemia". *Clinical biochemistry* 49:6, 2016, pp. 421–432.

244. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. "Striving for simplicity: The all convolutional net". In: *Workshop Track of the International Conference on Learning Representations (ICLR)*. 2015.

245. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". *Journal of Machine Learning Research* 15:1, 2014, pp. 1929–1958.

246. S. Stern. "State of the art in stress testing and ischaemia monitoring". *Cardiac electrophysiology review* 6:3, 2002, pp. 204–208.

247. J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay, and J. J. Collins. "A Deep Learning Approach to Antibiotic Discovery". *Cell* 180:4, 2020, 688–702.e13. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2020.01.021.

248. N. Strodthoff, P. Wagner, T. Schaeffter, and W. Samek. "Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL". *IEEE J. Biomed. Health Informatics* 25:5, 2021, pp. 1519–1528. DOI: 10.1109/JBHI.2020.3022989.

249. J. Suárez, S. García, and F. Herrera. "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges". *Neurocomputing* 425, 2021, pp. 300–322. DOI: 10.1016/j.neucom.2020.08.017.

250. A. Subasi. "EEG signal classification using wavelet feature extraction and a mixture of expert model". *Expert Systems with Applications* 32:4, 2007, pp. 1084–1093.

251. C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, et al. "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age". *PLoS medicine* 12:3, 2015, e1001779.

252. M. Sugiyama, F. Llinares-López, N. Kasenburg, and K. M. Borgwardt. "Significant Subgraph Mining with Multiple Testing Correction". In: *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*. Ed. by S. Venkatasubramanian and J. Ye. SIAM, 2015, pp. 37–45. DOI: 10.1137/1.9781611974010.5.

253. I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2014, pp. 3104–3112.

254. S. T. and P. B. Sivakumar. "Human Gait Recognition and Classification Using Time Series Shapelets". In: *2012 International Conference on Advances in Computing and Communications*. 2012, pp. 31–34. DOI: 10.1109/ICACC.2012.8.

255. Y. Tanaka, K. Iwamoto, and K. Uehara. "Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle". *Mach. Learn.* 58:2-3, 2005, pp. 269–300. DOI: 10.1007/s10994-005-5829-2.

256. Y. Tanglay, R. Twerenbold, G. Lee, M. Wagener, U. Honegger, C. Puelacher, T. Reichlin, S. M. Sou, S. Druey, T. Hochgruber, et al. "Incremental value of a single high-sensitivity cardiac troponin I measurement to rule out myocardial ischemia". *The American journal of medicine* 128:6, 2015, pp. 638–646.

257. R. E. Tarone. "A Modified Bonferroni Method for Discrete Data". *Biometrics* 46:2, 1990, pp. 515–522. DOI: 10.2307/2531456.

258. R. L. Teach and E. H. Shortliffe. "An analysis of physician attitudes regarding computer-based clinical consultation systems". *Computers and Biomedical Research* 14:6, 1981, pp. 542–558.

259. A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese. "Statistical significance of combinatorial regulations". *Proceedings of the National Academy of Sciences* 110:32, 2013, pp. 12996–13001.

260. N. Tishby and N. Zaslavsky. "Deep learning and the information bottleneck principle". In: *IEEE Information Theory Workshop (ITW)*. 2015, pp. 1–5.

261. A. Tong, J. Huang, G. Wolf, D. Van Dijk, and S. Krishnaswamy. "Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9526–9536.

262. A. Tong, G. Huguet, D. Shung, A. Natik, M. Kuchroo, G. Lajoie, G. Wolf, and S. Krishnaswamy. *Embedding Signals on Knowledge Graphs with Unbalanced Diffusion Earth Mover's Distance*. 2021. arXiv: 2107.12334 [cs.LG].

263. W. S. Torgerson. "Multidimensional scaling: I. Theory and method". *Psychometrika* 17:4, 1952, pp. 401–419.

264. J. Torres-Soto and E. A. Ashley. "Multi-task deep learning for cardiac rhythm detection in wearable devices". *NPJ digital medicine* 3:1, 2020, pp. 1–8.

265. C. D. Truong and D. T. Anh. "A Fast Method for Motif Discovery in Large Time Series Database under Dynamic Time Warping". In: *Knowledge and Systems Engineering - Proceedings of the Sixth International Conference KSE 2014, Hanoi, Vietnam, 9-11 October 2014*. Ed. by V. Nguyen, A. Le, and V. Huynh. Vol. 326. Advances in Intelligent Systems and Computing. Springer, 2014, pp. 155–167. DOI: 10.1007/978-3-319-11680-8_13.

266. M. Tsang, D. Cheng, and Y. Liu. "Detecting statistical interactions from neural network weights". In: *International Conference on Learning Representations (ICLR)*. 2018.

267. R. S. Tsay. *Analysis of financial time series*. Vol. 543. John wiley & sons, 2005.

268. K. Van den Berge, H. R. De Bezieux, K. Street, W. Saelens, R. Cannoodt, Y. Saeys, S. Dudoit, and L. Clement. "Trajectory-based differential expression analysis for single-cell sequencing data". *Nature communications* 11:1, 2020, pp. 1–13.

269. M. M. Van Den Ent, D. W. Brown, E. J. Hoekstra, A. Christie, and S. L. Cochi. "Measles mortality reduction contributes substantially to reduction of all cause mortality among children less than five years of age, 1990–2008". *The Journal of infectious diseases* 204:suppl_1, 2011, S18–S23.

270. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

271. H. J. Verberne, W. Acampa, C. Anagnostopoulos, J. Ballinger, F. Bengel, P. De Bondt, R. R. Buechel, A. Cuocolo, B. L. van Eck-Smit, A. Flotats, et al. "EANM procedural guidelines for radionuclide myocardial perfusion imaging with SPECT and SPECT/CT: 2015 revision". *European journal of nuclear medicine and molecular imaging* 42:12, 2015, pp. 1929–1940.

272. J.-P. Vert. *The optimal assignment kernel is not positive definite*. 2008. arXiv: 0801.4061 [cs.LG].

273. C. Villani. *Optimal transport: old and new*. Vol. 338. Springer, 2009.

274. J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, and L. G. Thijs. "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure". *Intensive Care Medicine* 22, 1996, pp. 707–710.

275. G. Wachman, R. Khardon, P. Protopapas, and C. R. Alcock. "Kernels for Periodic Time Series Arising in Astronomy". In: *Machine Learning and Knowledge Discovery in Databases*. Springer, Heidelberg, Germany, 2009, pp. 489–505.

276. C. Walck et al. "Hand-book on statistical distributions for experimentalists". *University of Stockholm* 10, 2007, p. 79.

277. J. E. Walter, U. Honegger, C. Puelacher, D. Mueller, M. Wagener, N. Schaerli, I. Strebel, R. Twerenbold, J. Boeddinghaus, T. Nestelberger, et al. "Prospective validation of a biomarker-based rule out strategy for functionally relevant coronary artery disease". *Clinical chemistry* 64:2, 2018, pp. 386–395.

278. Z. Wang, W. Ya, and T. Oates. "Time series classification from scratch with deep neural networks: A strong baseline". In: *International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 1578–1585.

279. S. Watanabe and H. Yamana. "Deep Neural Network Pruning Using Persistent Homology". In: *3rd IEEE International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2020, Laguna Hills, CA, USA, December 9-13, 2020*. IEEE, 2020, pp. 153–156. DOI: 10.1109/AIKE48582.2020.00030.

280. B. Weisfeiler and A. A. Lehman. "The reduction of a graph to canonical form and the algebra which appears therein". *Nauchno–Technicheskaja Informatsia* 9, 1968, pp. 12–16.

281. P. Werbos. "Backpropagation through time: what it does and how to do it". *Proceedings of the IEEE* 78:10, 1990, pp. 1550–1560. DOI: 10.1109/5.58337.

282. G. Wu, E. Y. Chang, and Z. Zhang. "An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines". In: *ICML*. 2005.

283. L. Wu, I. En-Hsu Yen, F. Xu, P. Ravikumar, and M. Witbrock. "D2KE: From Distance to Kernel and Embedding". *arXiv e-prints*, arXiv:1802.04956, 2018, arXiv:1802.04956. arXiv: 1802.04956 [stat.ML].

284. H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms". *arXiv preprint arXiv:1708.07747*, 2017.

285. J. Xu, Y. Zhang, P. Zhang, A. Mahmood, Y. Li, and S. Khatoon. "Data Mining on ICU Mortality Prediction Using Early Temporal Data: A Survey". *Int. J. Inf. Technol. Decis. Mak.* 16:1, 2017, pp. 117–160. DOI: 10.1142/S0219622016300020.

286. Y. Xu, F. Sun, and X. Zhang. "Literature survey of active learning in multimedia annotation and retrieval". In: *International Conference on Internet Multimedia Computing and Service, ICIMCS '13, Huangshan, China - August 17 - 19, 2013*. Ed. by K. Lu, T. Mei, and X. Wu. ACM, 2013, pp. 237–242. DOI: 10.1145/2499788.2499794.

287. L. Ye and E. Keogh. "Time Series Shapelets: A New Primitive for Data Mining". In: *KDD*. ACM, New York, NY, USA, 2009, pp. 947–956.

288. L. Ye and E. J. Keogh. "Time series shapelets: a new primitive for data mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*. Ed. by J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki. ACM, 2009, pp. 947–956. DOI: 10.1145/1557019.1557122.

289. Y. Ying, C. Campbell, and M. Girolami. "Analysis of SVM with Indefinite Kernels". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 22. Curran Associates, Inc., 2009, pp. 2205–2213.

290. I. T. Au-Yong, N. Thorn, R. Ganatra, A. C. Perkins, and M. E. Symonds. "Brown adipose tissue and seasonal variation in humans". *Diabetes* 58:11, 2009, pp. 2583–2587.

291. J. You, B. Liu, Z. Ying, V. S. Pande, and J. Leskovec. "Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 6412–6422. URL: https://proceedings.neurips.cc/paper/2018/hash/d60678e8f2ba9c540798ebbde31177e8-Abstract.html.

292. M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision (ECCV)*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8689. Lecture Notes in Computer Science. Springer. Cham, Switzerland, 2014, pp. 818–833.

293. S. Zeng, F. Graf, C. D. Hofer, and R. Kwitt. "Topological Attention for Time Series Forecasting". *CoRR* abs/2107.09031, 2021. arXiv: 2107.09031. URL: https://arxiv.org/abs/2107.09031.

294. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Sy8gdB9xx.

295. J. Zhang, X. Li, L. Gao, L. Wen, and G. Liu. "A Shapelet Dictionary Learning Algorithm for Time Series Classification". In: *15th IEEE International Conference on Automation Science and Engineering, CASE 2019, Vancouver, BC, Canada, August 22-26, 2019*. IEEE, 2019, pp. 299–304. DOI: 10.1109/COASE.2019.8843231.

296. M. Zhang, Q. Su, Y. Lu, M. Zhao, and B. Niu. "Application of machine learning approaches for protein-protein interactions prediction". *Medicinal Chemistry* 13:6, 2017, pp. 506–514.

297. S. Zhou, E. Zelikman, F. Lu, A. Y. Ng, G. E. Carlsson, and S. Ermon. "Evaluating the Disentanglement of Deep Generative Models through Manifold Topology". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=djwS0m4Ft%5C_A.

298. Y. Zhu, C. M. Yeh, Z. Zimmerman, K. Kamgar, and E. J. Keogh. "Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds". In: *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 837–846. DOI: 10.1109/ICDM.2018.00099.

299. Y. Zhu, Z. Zimmerman, N. S. Senobari, C. M. Yeh, G. J. Funning, A. Mueen, P. Brisk, and E. J. Keogh. "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins". In: *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*. Ed. by F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu. IEEE Computer Society, 2016, pp. 739–748. DOI: 10.1109/ICDM.2016.0085.