DISS. ETH NO. 28104

# Machine Learning on Clinical Time Series: Classification and Representation Learning

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by

Michael Moor
Doctor of Medicine of the University of Basel

born on 30.01.1992
citizen of Hasliberg, Bern, Switzerland

examined by

Prof. Dr. Karsten Borgwardt, examiner
Prof. Dr. Smita Krishnaswamy, co-examiner
Prof. Dr. Jenna Wiens, co-examiner

2022

*In memoriam patris mei.*

# ABSTRACT

The life sciences of the digital era are driven by its most fundamental and irreplaceable currency: data. The advent of big data and machine learning (ML) algorithms has promised to revolutionise biomedical sciences and medical practice by means of automated diagnostics, data-driven disease subtyping and personalised treatments. However, while ML in health has become a vibrant field, in many cases the translation into practice has turned out to be more challenging than expected, or to put it more bluntly: the revolution is still pending. In this dissertation, we identify a set of challenges that arise when trying to leverage ML on clinical data, specifically for time series classification problems. Even though raw patient data are now being routinely collected in unprecedented amounts of electronic health records, typically, this data first needs to be carefully curated, preprocessed and annotated in order to arrive at a dataset that may be used in a ML pipeline to solve a down-stream prediction problem. Due to the complexity of this process already for a single dataset, external validations—albeit crucial—are frequently missing in existing studies.

In the first part of this thesis, we consider the classification of clinical time series, in particular the application domain of sepsis prediction, where the goal is to early detect sepsis, a potentially fatal complication to infections. We propose mitigation strategies to the aforementioned issues by creating a large, multi-centric cohort of intensive care unit (ICU) patients with temporally annotated sepsis labels. This allowed us to perform the first *international* development and validation of sepsis prediction models using ML. Along the way, we found that federated learning and model sharing (as opposed to data sharing) leads to convincing performance—without requiring to physically export sensitive patient data outside the source site. Moreover, we encountered clinical time series of vital and laboratory measurements that were irregularly spaced and, for a given time step, incompletely observed. Throughout this thesis, we addressed informative missingness of data using Gaussian process models.

After an application-focused first part, the second part of this dissertation considers the model's inner workings more closely. Starting with irregularly sampled time series, we investigate path signatures, a powerful transform (that can be used as a neural network layer) to encode paths of data at virtually no loss of information. In particular, we explore how these signatures may be used to learn time series representations that lead to beneficial classification performance. We thereby uncover that the way the signature "interprets" raw data has drastic implications that are reflected in down-stream performance. We then propose a novel variant of Gaussian process adapters that lead to more robustness in signature-based models.

Finally, after having considered model's implicit interpretation of data, in the final chapter, we explore how models can learn and preserve structures that are available in the raw

(and potentially high-dimensional) input data. For this, we leverage concepts from topological data analysis, and propose topological autoencoders, a novel deep learning architecture that can preserve complex structures and shapes of intangibly high-dimensional data in low-dimensional visualisations. In summary, we hope that our contributions to clinical time series classification will pave the way for the deployment of robust and validated models that create clinical value for the monitored patients. Moreover, we envision that our findings in temporal and topological representation learning will illuminate the analysis and understanding of the ever more accumulating wealth of large and high-dimensional biomedical datasets.

# ZUSAMMENFASSUNG

Die Lebenswissenschaften des digitalen Zeitalters werden von einer fundamentalen und unersetzlichen Grundwährung vorangetrieben: den Daten. Die Erhebung immenser Mengen an digitalen Daten und dessen Analyse mit neuen Methoden des maschinellen Lernens (ML) versprach einen grundlegenden Paradigmenwechsel der biomedizinischen Wissenschaften als auch der medizinischen Praxis. Dies mittels Automatisierungen in der Diagnostik, Subtypisierungen von Krankheiten und schliesslich personalisierten Behandlungen. Obschon die Anwendung von ML in der Biomedizin und dem Gesundheitswesen zu einem aktiven Forschungsfeld heranwuchs, haben verschiedene Tücken eine breite praktische Umsetzung dessen Einsichten herausgezögert, oder kurzum: eine Revolution ist bisher noch ausgeblieben.

In dieser Dissertation zeigen wir eine Reihe von Herausforderungen auf, die sich ergeben, wenn man versucht, das maschinelle Lernen (ML) auf klinische Daten anzuwenden, insbesondere bei Problemen der Zeitreihenklassifizierung. Obwohl Patientenrohdaten heute routinemässig in noch nie dagewesenen Mengen in elektronischen Patientendossiers gesammelt werden, müssen diese Daten in der Regel zunächst sorgfältig überprüft, kuratiert, vorverarbeitet und annotiert werden, um einen Datensatz zu erhalten, der in einer maschinellen Lernumgebung zur Lösung eines nachgelagerten Vorhersageproblems verwendet werden kann. Aufgrund der Komplexität dieses Prozesses bereits für einen einzelnen Datensatz, fehlen in vielen bestehenden Studien externe Validierungen, obwohl diese von entscheidender Bedeutung wären.

Im ersten Teil dieser Arbeit befassen wir uns mit der Klassifizierung klinischer Zeitreihen, insbesondere mit der Anwendungsdomäne der Sepsisvorhersage, bei der es darum geht, Sepsis, eine potenziell tödliche Komplikation von Infektionen, frühzeitig zu erkennen. Wir schlagen Strategien zur Entschärfung der oben genannten Probleme vor, indem wir eine grosse, multizentrische Kohorte von Intensivpatienten mit zeitlich aufgelösten Sepsis-Annotationen erstellen. Dies ermöglichte uns die erste internationale Entwicklung und Validierung von Sepsis-Vorhersagemodellen durch maschinelles Lernen. Dabei haben wir festgestellt, dass föderiertes Lernen und der Transfer von Modellen (im Gegensatz zum direkten Datentransfer zwischen Spitälern) zu einer überzeugenden Sepsis-Früherkennung führen—ohne dass sensible Patientendaten die vier Wände des Spitals verlassen müssen, in welchem die Daten ursprünglich erhoben wurden. Im Laufe dieser Arbeit stiessen wir auf klinische Zeitreihen von Vital- und Labormessungen, die in unregelmässigen Abständen und für jeweilige Zeitschritte unvollständig beobachtet wurden. Dieses informative Fehlen von Daten haben wir mit Hilfe von Gauß Prozessen modelliert und berücksichtigt .

Nach einem anwendungsorientierten ersten Teil wird im zweiten Teil dieser Dissertation die innere Funktionsweise der untersuchten Modelle (neuronalen Netze) näher betrachtet. Ausgehend von unregelmässig beobachteten Zeitreihen untersuchen wir die Pfadsignatur, eine mächtige Transformation (die als Baustein in neuronalen Netzen verwendet werden kann), um Pfade im Datenraum praktisch ohne Informationsverlust zu kodieren, was interessante Anwendungen für Zeitreihen birgt. Insbesondere untersuchen wir, wie diese Signaturen verwendet werden können, um Zeitreihenrepräsentationen zu lernen, die sich vorteilhaft auf eine nachgeschaltete Zeitreihenklassifizierung auswirkt. Dabei stellen wir fest, dass die Art und Weise, wie die Signatur die Rohdaten "interpretiert", drastische Auswirkungen hat, die sich in der Güte der Klassifikation wiederspiegeln. Anschliessend schlagen wir eine neuen Ansatz von Gauß'schen Prozess Modellen vor, die zu mehr Robustheit in signaturbasierten Modellen führt.

Nachdem wir uns mit der impliziten Interpretation von Daten durch ML Modelle beschäftigt haben, untersuchen wir im letzten Kapitel, wie Modelle Strukturen lernen und in internen Datenrepräsentationen bewahren können, die in den rohen (und potenziell hochdimensionalen) Eingabedaten vorhanden sind. Dazu verwenden wir Konzepte aus der topologischen Datenanalyse und entwickeln "Topological Autoencoders", eine neue Modell-Architektur, die komplexe Strukturen und Formen von hochdimensionalen Datenräumen in niedrigdimensionalen Visualisierungen bewahren kann. Zusammenfassend hoffen wir, dass unsere Beiträge zur Klassifizierung klinischer Zeitreihen den Weg für den Einsatz robuster und validierter ML Modelle ebnen werden, die einen klinischen Nutzen für die untersuchten Patienten hervorbringen. Darüber hinaus erhoffen wir, dass unsere Erkenntnisse im Bereich des Lernens von zeitlichen und topologischen Repräsentationen die Analyse und das Verständnis der heranwachsenden Fülle an hochdimensionalen biomedizinischen Datensätzen verbessern werden.

# ACKNOWLEDGEMENTS

I would like to express my solemn gratitude to my advisor, Prof. Dr. Karsten Borgwardt, who over the years provided me with invaluable supervision, continuous support, countless academic opportunities, and fruitful discussions. He created a welcoming work environment that made it possible for me to pursue long-term research projects, while also empowering me to acquire novel skills and to develop and realise my own research agenda. Furthermore, he kindly provided the ressources that allowed me to attend conferences to learn from and engage with the international research community. Finally, I wish to explicitly thank him for the consideration, guidance, and patience with which he met and supported my transition to fatherhood during the doctoral studies.

Next, I would like to extend my gratitude to Prof. Dr. Smita Krishnaswamy and Prof. Dr. Jenna Wiens, who kindly agreed to participate in the thesis examination committee, as well as to Prof. Dr. Jörg Stelling for chairing the doctoral examination.

I would like to deeply thank all the collaborators and co-authors with whom I had the pleasure to work together during my doctoral studies. My very special gratitude goes to Dr. Bastian Rieck who over the span of my studies has relentlessy supported, supervised, and actively engaged with my work. As an invaluable mentor and role model, he has taught me how to take a more mathematical view to scrutinise concepts in machine learning, and introduced me to the marvelous world of topological data analysis. Furthermore, I would like to express my deep gratitude to Dr. Damián Roqueiro, who during my first year of doctoral studies provided valuable feedback and guided me through my first project.

My deepest thank goes to Max Horn and Christian Bock, with whom I had the chance to closely collaborate on several fruitful projects some of which resulted in shared first authorships. Many skills that I have acquired during my studies, I feel are at least partially owed to my glorious office members, Max, Christian, and Bastian, who provided me with an inexhaustible supply of profound discussions, late-night coding sessions, and lastly, sincere friendship.

Fhistheremore, I would like to express my gratitute to Matteo Togninalli, Thomas Gumbsch, Catherine Jutzeler, Juliane Klatt, Bowen Fan, Lucie Bourguignon, and Dr. Felix Hensel, with whom I had the pleasure to collaborate on several exciting projects. I am also deeply thankful to the current and past members of Prof. Karsten Borgwardt's lab, including Dr. Katharina Heinrich, Caroline Weis, Giulia Muzio, Tim Kucera, Leslie O'Bray, Dr. Sarah Brüenigk, Dr. Dexiong Chen, Dr. Carlos Oliver, Dr. Michael Adamer, Dr. Elisabetta Ghisu, Dr. Anja Gumpinger, Dr. Lukas Folkman, Dr. Laetitia Papaxanthos, and Dr. Felipe Llinares-López, for being kind and inspiring colleagues.

I would also like to offer my gratitude to several external collaborators that enriched several projects throughout my doctoral studies with expertise from adjacent disciplines and invaluable outside perspectives. This includes Nicolas Bennett, Drago Plečko, Edward De Brouwer, Prof. Dr. Adrian Egli, Prof. Dr. Peter Bühlmann, Zhiliang Wu, Jannis Born, and Patrick Kidger. Moreover, I would like to thank the department's staff, in particular Cindy Malnasi for having provided excellent administrative support.

Finally, I am extremely grateful to my family and friends for standing by my side through the many ups and downs of this adventure. In particular, I am deeply grateful to my father, Dr. Hans Moor, whose fascination for science and nature has sparked my scientific curiosity from an early age on. Next, I am deeply thankful to my mother, Charlotte Moor, who has been of immense support throughout my entire education, and in particular during the last phase of my doctoral studies.

Next, I would like to express my gratitude to my four older brothers, Prof. Dr. Andreas Moor, Dr. Matthias Moor, Thomas Moor, and Markus Moor, who have been my oldest friends, and who inspired and shaped the very person I now have become.

Lastly, I am immeasurably indebted to my incredible wife, Dr. Manuela Moor, whose encouragement, love, support, and wisdom guided me to steer through even the roughest waters on this journey. To her, I owe two wonderful daughters, Lilith and Mara, who have been brightening and enriching my life in previously unimaginable ways ever since.

# CONTENTS

1 INTRODUCTION 1

1.1 Clinical time series classification . . . . . . . . . . . . . . . . . . . . . . . . 6

1.1.1 Uncertainty-aware recognition of sepsis with Gaussian Process Temporal Convolutional Networks . . . . . . . . . . . . . . . . . . . . . . 6

1.1.2 Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.2 Temporal and topological representation learning . . . . . . . . . . . . . . . 7

1.2.1 Path signatures for time series representation learning . . . . . . . 8

1.2.2 Topological representation learning . . . . . . . . . . . . . . . . . . 9

1.3 Organisation of the thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

I CLINICAL TIME SERIES CLASSIFICATION 15

2 PROBLEM FORMULATION 17

2.1 Time series classification . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.1.1 What is a time series? . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.1.2 Time series notation . . . . . . . . . . . . . . . . . . . . . . . . . . 18

2.1.3 Time series prediction tasks . . . . . . . . . . . . . . . . . . . . . . 20

2.1.4 Methods for time series classification . . . . . . . . . . . . . . . . . 23

2.2 Early prediction of sepsis . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2.2.1 What is sepsis? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

2.2.2 Related work on the early prediction of sepsis . . . . . . . . . . . . 26

3 UNCERTAINTY-AWARE RECOGNITION OF SEPSIS WITH GAUSSIAN PROCESS TEMPORAL CONVOLUTIONAL NETWORKS 29

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.2 Gaussian processes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

3.2.1 Gaussian process regression . . . . . . . . . . . . . . . . . . . . . . 31

3.2.2 Multi-task Gaussian processes . . . . . . . . . . . . . . . . . . . . . 33

3.2.3 Gaussian processes adapters . . . . . . . . . . . . . . . . . . . . . 35
3.3 Temporal convolutional networks . . . . . . . . . . . . . . . . . . . . . . . 37
3.4 MGP-TCN: Gaussian Process Temporal Convolutional Networks . . . . . . 38
3.5 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
3.5.1 Dataset and sepsis label . . . . . . . . . . . . . . . . . . . . . 40
3.5.2 Prediction problem . . . . . . . . . . . . . . . . . . . . . . . . 42
3.5.3 Data filtering . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.5.4 Comparison methods . . . . . . . . . . . . . . . . . . . . . . . 45
3.5.5 Experimental setup . . . . . . . . . . . . . . . . . . . . . . . . 46
3.5.6 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
3.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

4 PREDICTING SEPSIS IN MULTI-SITE, MULTI-NATIONAL INTENSIVE CARE COHORTS USING DEEP LEARNING 53
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.2 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.2.1 Study design and data sources . . . . . . . . . . . . . . . . . . . 55
4.2.2 Outcome and prediction problem . . . . . . . . . . . . . . . . . . 60
4.2.3 Prediction methods . . . . . . . . . . . . . . . . . . . . . . . . 64
4.2.4 Feature engineering . . . . . . . . . . . . . . . . . . . . . . . . 67
4.2.5 Experimental setup . . . . . . . . . . . . . . . . . . . . . . . . 68
4.2.6 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
4.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
4.3.1 Dataset characteristics . . . . . . . . . . . . . . . . . . . . . . . 71
4.3.2 Internal validation . . . . . . . . . . . . . . . . . . . . . . . . . 73
4.3.3 External validation . . . . . . . . . . . . . . . . . . . . . . . . 73
4.3.4 Model inspection . . . . . . . . . . . . . . . . . . . . . . . . . 78
4.3.5 Calibration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
4.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85

II TEMPORAL AND TOPOLOGICAL REPRESENTATION LEARNING 87

5 PATH SIGNATURES FOR TIME SERIES REPRESENTATION LEARNING 89
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89

5.2 The path signature . . . . . . . . . . 90
    5.2.1 Properties of the signature . . . . . . . . . . 92
    5.2.2 Computing the signature . . . . . . . . . . 94
5.3 Path imputation of signature models . . . . . . . . . . 95
5.4 Related work . . . . . . . . . . 98
5.5 Experiments . . . . . . . . . . 100
5.6 Discussion . . . . . . . . . . 104
5.7 Concluding remarks . . . . . . . . . . 105

6 TOPOLOGICAL REPRESENTATION LEARNING 107
6.1 Introduction . . . . . . . . . . 108
6.2 What is topology? . . . . . . . . . . 108
6.3 Persistent homology: topology at multiple scales . . . . . . . . . . 111
6.4 A topology-preserving autoencoder . . . . . . . . . . 114
    6.4.1 Topological autoencoder . . . . . . . . . . 115
    6.4.2 Stability . . . . . . . . . . 118
    6.4.3 Scalability and runtime . . . . . . . . . . 122
6.5 Related work . . . . . . . . . . 122
6.6 Experiments . . . . . . . . . . 123
    6.6.1 Experimental setup . . . . . . . . . . 123
    6.6.2 Results . . . . . . . . . . 125
6.7 Discussion . . . . . . . . . . 126

7 CONCLUSION 133
7.1 Part I: Clinical time series classification . . . . . . . . . . 133
    7.1.1 Uncertainty-aware recognition of sepsis with Gaussian Process Temporal Convolutional Networks . . . . . . . . . . 135
    7.1.2 Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning . . . . . . . . . . 136
7.2 Part II: Temporal and topological representation learning . . . . . . . . . . 138
    7.2.1 Path signatures for time series representation learning . . . . . . . . . . 138
    7.2.2 Topological representation learning . . . . . . . . . . 140
7.3 Outlook . . . . . . . . . . 140

SUPPLEMENTARY MATERIALS 145
A.1 Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning . . . 145
A.1.1 Results on the Emory dataset . . . 145
A.2 Path signatures for time series representation learning . . . 162
A.3 Topological representation learning . . . 165
ACRONYMS 167
BIBLIOGRAPHY 169

# 1 INTRODUCTION

Over the last three centuries, the practice of medicine has increasingly become a scientific discipline based on empirical evidence, rather than an art as it had been before [49]. The last century has then witnessed the birth of modern, evidence-based medicine as we know it today. In essence, this means that we require knowledge about diseases—how to classify them, how to diagnose and treat them, and how best provide care for the patients that are plagued by them—to be backed by supporting *data*. Today, we may immediately associate keywords like "data", or "data-driven" with computers, algorithms, databases, tech companies, the world-wide web, and so on. But the biomedical disciplines, and indeed natural sciences in general, have been data-driven or "data-centric" long before personal computers found their way into our private homes, our workplaces, or even our pockets. In the pre-digital age, data was typically stored in manually registered records, where its creation depended on the rare skill of literacy and it usability was fully at the mercy of the author's handwriting and of the physical integrity of the surrounding storage space [157]. In contrast, digitalised data are scalable[1], mobile, reusable, and offer a wealth of new opportunities.

Coming back to the 21st century, an on-going digital revolution keeps pulsating waves of new technologies that are permeating hospitals, research institutions, and medical practises, thereby redefining the very essence of what it means to provide patient care. Posterity may think of the medicine of the preceding century as the one that popularised an evidence-based medicine, where for instance large scale clinical trials enabled the discovery of medical knowledge that goes beyond anecdotal expert opinions. In contrast, the current era may likely be shaped and remembered for the wide-reaching digital transformation that creates numerous ramifications throughout medicine by essentially redefining how we collect and store medical information, how we subtype existing (or discover novel) diseases, and how we discover or even synthesise remedies to illnesses that previously were believed to be incurable [198].

Over the last years, an ever more digital and computerised biomedicine has lead to significant breakthroughs. For instance, in a pre-digital age it would have been absolutely unthinkable to develop, carefully test, and roll out highly effective vaccines, thereby supplying

---

[1] For instance, we may fit the contents of an entire library on a single USB stick.

almost 4 billion people with at least one dose, less than *two years* into a global pandemic, that was caused by a newly emerged respiratory virus [91]. Nevertheless, the digitalisation of medicine has brought with it certain pitfalls, that are holding back progress and delay successful translations and deployments of new digital solutions and algorithms that may improve patient care. For instance, a lack of data interoperability between institutions—or even between the departments within an institution—can make the exchange and analysis of health record data cumbersome [12]. Furthermore, current health information technology (IT) systems may increase clinicians' burden of stress and even lead to burnout [64]. Additionally, while it used to be straight-forward to physically lock a room filled with sensitive patient records, ensuring the safety of digital patient data is less obvious and nowadays poses a significant technical challenge and a potential legal hazard for any health care provider. This issue is further exacerbated with the *sharing* of patient data which is encouraged by the trend to collect ever larger, multi-centric datasets in the spirit of creating more globally representative cohorts.

Even though a wealth of high-resolution and multi-modal patient data is being routinely collected and digitally stored across hospital wards and high-tech intensive care units (ICUs), a large portion of this valuable data is not analysed further—beyond the scope of immediate patient care. In ICUs, for example, caught in a continuous and almost overwhelming stream of patient monitoring data, clinicians are challenged to identify which patients need immediate attention, which available measurements are relevant, which data still need to be collected, and how to proceed with the patient management.

We are living the futuristic times where even our wrist watches are smart enough to tell us when to eat, or when to do sports, or where our entertainment system predicts the next movie we would enjoy watching. One could therefore be tempted to assume that also when entering a hospital, behind the curtains a smart and data-driven IT system would support data-overwhelmed clinicians, by efficiently orchestrating the relevant flow of information and by making evidence-based recommendations and predictions. In practice, however, clinical medicine tends to limp behind other work sectors, technology-wise, which Topol [198] referred to as *shallow medicine*. There are numerous efforts to integrate algorithms into everyday practice, for instance to raise alarms for unstable patients [81][2] or to automatically classify fundus images [127]. However, in non-academic institutions, an admitted patient may not be treated with more algorithmic sophistication (in terms of automated predictions) than an alarm that triggers if certain vital parameters (like mean arterial blood pressure) pass a threshold of predefined lower and upper bounds [174].

---

[2] Notably, the here deployed early warning score is rule-based, on not entirely data-driven.

So why are we lagging behind in the medical equivalent of otherwise widely adopted recommender systems, e.g. in the form of clinical decision support systems? To grasp this question, we first need to understand how the underlying predictive computational algorithms work. Computational algorithms that learn to make predictions about a target of interest by means of mining and exploiting correlations in large datasets can be subsumed under the umbrella term *machine learning*. Irrespective of the nature of the prediction target—be it a discrete class (like “cancerous lesion” versus “healthy tissue” in digital pathology), or a continuous target (like estimated time of survival)—typically, a machine learning (ML) model is trained on a large amount of data samples in order to learn to make predictions about previously unseen data. Depending on the application case, data samples are paired with labels, and then the goal is learning to predict the corresponding label of an new data point. This scenario is referred to as supervised learning. This can be juxtaposed with unsupervised learning, where label information is not available during training, and where to goal is predict targets which are not predefined upon training time (e.g. clustering, or dimensionality reduction).

It may have become straight-forward to train a ML system to detect and classify objects in natural images [220], or to process and translate natural languages from text data [85]. However, to successfully leverage ML for medical prediction tasks still offers a distinct set of challenges that may explain the years (if not decades) of delay we observe in ML solutions being deployed in clinical workflows.

DATASETS ML models excel in scenarios where data is abundant. By crawling the internet, it has become possible to create enormous datasets comprising millions of images, videos, text documents and so on. In contrast, the creation of large biomedical datasets is considerably more delicate. For instance, to compile an electronic health record (EHR) dataset, we need a data model, i.e., a concept on how to encode and store information in a way that it can be efficiently queried and linked to other data. But before we can even start thinking about the complexities invoked by health record data, we first need to get *access* to these data, which remains a key limiting factor for the analysis of EHRs. Recent efforts to make deidentified patient data publicly available for research has significantly facilitated ML studies on these types of data [98]. However, the creation of datasets such as the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III) dataset is a tremendous effort, and the sharing of sensitive patient data is inherently more hazardous than other non-sensitive types of data that are regularly used in ML.

TASKS AND UTILITY Even when a large database of patient data is accessible, we cannot directly train a model on these data, but need to carefully consider several viewpoints. To list a few:

1. Which target do we aim to predict? For instance, patient mortality, length of hospital stay, adverse reactions to treatments, patient deterioration, readmission after discharge, etc.
2. What entity refers to a data sample? To give an example, are we making predictions about a specific culture sample (e.g., will there be an antibiotic resistance, or not?), or are we predicting the outcome of a patient based on the admission state, or based on sliding or expanding time windows of monitoring data?
3. How can a given prediction be useful, and who benefits from it? E.g., will it benefit patients, will it support clinicians, or will it facilitate insurances in monitoring patients?

This enumeration illustrates that there are various study design choices to be made, and not many of them will lead to a clinically useful prediction output. Sometimes, the tasks that can be defined and implemented most easily (e.g. predicting patient mortality or predicting a billing code related to a disease) may not be the ones that will ultimately lead to clinically useful insights or actionable warnings.

LABELS Now, given a dataset and a prediction task, a next challenge is to obtain ground-truth labels. For instance, were we to analyse thoracic X-ray images in order to detect abnormalities (pneumonia, cancerous lesions and more), a typical approach would require manual annotations of each image by a radiological expert. However, expert annotations of patient data are usually labour-intense and come at high costs [214]. The scarcity of high-quality labels has been a core topic in ML research well beyond health applications [222]. But especially in health applications, were predictions are generally required to be reliable, the lack of high-quality labels remains a core hindrance.

VALIDITY As a next obstacle for clinical applications of ML, data distributions can differ drastically between centres and even change within the same centre over time, or between retrospective and prospective data collections [153]. Reasons for such distribution shifts are manifold. They may be rooted in different devices that record data, different policies (that lead to distinct diagnostics and therapeutics), or just different cohorts with distinct characteristics. In any case, heterogeneity within and between patient datasets pose a key problem for ML models that are tasked to learn patterns and signals that allow for predictions that are generalisable to previously unseen data.

Missingness A next challenge, that frequently arises amongst various types of patient data, is data *missingness*. For example, in longitudinal clinical data, time series of laboratory measurements are only sparsely observed, where the missingness itself can be informative [176]. To properly handle missing data has become an entire subdiscipline of statistics with a broad area of applications [66, Chapter 25]. Furthermore, how exactly we account for missing values, carries implicit information about how we interpret both the underlying data generating process as well as the way we observe a discrete set of measurements from it.

We now have enumerated a battery of problems that illustrate why it is challenging to properly organise and process patient data, and that this is necessary for leveraging ML to obtain clinically useful predictions. In this dissertation, we aim to explore and by part address these open problems. From a high-level perspective, in this thesis we consider three steps of the ML pipeline that can be sequentially arranged:

$$\text{data} \rightarrow \text{model} \rightarrow \text{predictions}.$$

Over the chapters of this thesis, our focus traverses these concepts, from back to front. We start out by attending to model *outputs* in the application context of clinical prediction models. Next, we focus on internal states of the model, i.e., learned representations and how they may effect downstream predictive performance. Lastly, we arrive at the front of the above paradigm and explore a class of models that are capable of capturing and preserving properties and structures of the input data in order to learn faithful representations and visualisations. Subsequently, we give more low-level details about the individual challenges to be addressed.

The first part of the thesis focuses on clinical time series classification. Here, we consider the challenges of arriving at a clinically meaningful prediction task, to create large-scale annotated datasets which in turn allow for the elaborated tasks to be solved using ML. In this context, we further explore how missingness information may be leveraged and how we can learn to generalise to new datasets despite distribution shifts. In a second part, zooming away from model outputs (clinical predictions, predictions under distribution shift), we attend to the internal representations that are learned by ML models and observe that implicit modelling choices not only determine missing data handling, but also affect the way models handle unobserved continuous data processes, more generally. Finally, we explore a scenario where learned representations may not be required to maximise class separability, but where the aim is to merely reduce the dimension of high-dimensional data while preserving shapes and structures that are present in the intangible data space. Having outlined a rough blue print of this thesis, we now briefly introduce and motivate both parts and their respective chapters.

## 1.1 CLINICAL TIME SERIES CLASSIFICATION

Since before the dawn of the digital age, medical practitioners have been challenged with the intricate task to observe and care for patients. Clinical practice may nowadays be spiked with a plethora of technical devices that measure and collect a multifarious range of parameters and data modalities. As if straight out of a science-fiction novel, we employ machines to count and sort our cells [69], to conjure actual annihilation events (via PET scanners) [8], and to produce images of living organs at the resolution of cellular structures [207]. But behind all this high-tech equipment, teams of doctors and nurses are observing, interacting with, and providing care to patients. Inherently, their task is of a *temporal* nature. This holds true both for the intensive care unit (ICU) specialist who continuously monitors the mean arterial pressure of her patient that went into shock, as well as for the general physician that follows up with his patient to assess whether the prescribed antidepressants showed an effect over the last month.

When taking a patient- and clinician-focused perspective, even in a modern and digitalised medicine, the temporal component remains essential. For comparison, imaging disciplines such as radiology or pathology have been curating and analysing digital imaging records using machine learning (ML) for decades. In contrast, even though the collection of temporally resolved electronic health records (EHRs) has been going on for decades, the curation and analysis of this complex, unstructured, noisy, and increasingly multi-modal data presupposes layers of preprocessing [148] which altogether has grown into a core discipline among digital health research. It is this type of data, clinical time series from EHRs, that we consider in the first part of this dissertation. Specifically, we focus on the data-rich environment of intensive care units (ICUs), and formulate an early warning problem, namely the detection of *sepsis*. Using this running application example, we consider several points of the aforementioned list of challenges. Before diving into these chapters, Chapter 2 gives a more in-depth introduction to Part I by introducing time series, providing some basic notation and defining the application domain, sepsis.

### 1.1.1 UNCERTAINTY-AWARE RECOGNITION OF SEPSIS WITH GAUSSIAN PROCESS TEMPORAL CONVOLUTIONAL NETWORKS

In Chapter 3, we conduct a pilot study for the early prediction of sepsis, a potentially fatal condition that describes a dysregulated host response to infection [185] (see Section 2.2.1).

For this, we created the first publicly available sepsis dataset equipped with hourly labels. We then propose a new model for sepsis prediction which in an end-to-end differentiable fashion combines uncertainty-awareness of Gaussian processes with dilated causal convolutions, i.e., temporal convolutional networks (TCNs), which exhibit a powerful inductive bias for temporal data [7]. In this study, we conduct a retrospective single-centre analysis and demonstrate that our proposed method, MGP-TCN exhibits beneficial predictive performance compared to several baselines.

### 1.1.2 PREDICTING SEPSIS IN MULTI-SITE, MULTI-NATIONAL INTENSIVE CARE COHORTS USING DEEP LEARNING

Chapter 4 can be seen as the logical consequence of the insights we took from Chapter 3, i.e. the corresponding sepsis prediction study [142], as well as our subsequent systematic review of sepsis prediction [143]. In this review, we found that the vast majority of sepsis prediction studies were not externally validated. In fact, they *could* not be validated, since access to publicly available sepsis datasets—other than the MIMIC-III dataset (which was already used in roughly half of the included studies)—was lacking. This rather dicey state of the literature motivated the study we present in this chapter. Our goal was to collect and harmonise a large, multi-centric intensive care unit (ICU) cohort, in order to develop and externally validate sepsis prediction approaches. By making this cohort publicly available[3], we aimed to facilitate further validations of sepsis prediction models by the research community—which up until now were practically impossible to externally validate due to the lack of public and annotated datasets. Furthermore, we address another limitation of the study in Chapter 3, namely that the considered models were not directly applicable in online prediction scenarios, which any early warning system ultimately would face upon deployment. In this multi-centre study, we address this issue by carefully formulating the prediction task as an online prediction problem, and even confront the included models with an online learning scenario during *training*. We then conduct an exhaustive internal and external validation and showcase a federated learning strategy to overcome dataset and label shifts upon external testing.

## 1.2 TEMPORAL AND TOPOLOGICAL REPRESENTATION LEARNING

During the second half of the last century, machine learning (ML) has emerged as a subdiscipline of computer science and statistics that may be interpreted as the marriage of scalable computing algorithms with statistical learning. While over the last decades a large bouquet

---

[3] The code used in this study to create the cohorts and conduct the analyses will be made public upon publication.

of ML techniques have been developed, at the core of any ML pipeline is essentially the aim to make *predictions* based on a set of input data. This framework is quite generic in that this input data may describe any kind of objects: for instance, natural images of cats and dogs, users in a social media application, patients being monitored in a hospital, and so on. Conventional ML approaches compute predictions about the data objects of interest by means of *features* that quantify and encode properties of the data objects. For instance, were we to predict if a hospitalised patient will survive her stay, we could encode the patient's state as a vector that accounts for various information that could be relevant for making the desired prediction, be it the time since admission, the patient's age, measurements of vital signs, existing comorbidities, etc. Or, if we want to detect a cat in a greyscale image, we could compute various image descriptors such as local summary statistics, texture features or edge maps and evaluate if based on these hand-crafted features, cats may indeed be detected. This process of encoding data objects (that itself may be intangible) into numeric feature vectors that can be used in a downstream learning algorithm, is referred to as *feature engineering*.

While in classical ML, feature encodings (or data representations) were typically manually engineered and then kept fixed during the learning process, with the era of deep learning the concept of automated feature learning, or *representation learning* has been popularised [10]. In contrast to a setting where we learn to make predictions based on a set of hand-crafted features, here, the new task is to learn both the feature representations, as well as the predictions based on said features, jointly.

The automated learning of representations has been made possible with the introduction of end-to-end differentiable learning algorithms by means of the backpropagation algorithm, which was popularised in the late 1980s [177][4]. In an in-depth review, Bengio et al. [10] elaborated on this concept, exploring what makes up for "good" learned representations, and put this idea into a larger context, subsuming elements from probabilistic modelling, dimensionality reduction and manifold learning. In the two chapters of Part II, we take two perspectives on representation learning, which we briefly sketch in the following sections.

### 1.2.1 PATH SIGNATURES FOR TIME SERIES REPRESENTATION LEARNING

In Chapter 5, we focus on the learning of representations of temporal data, and equip ourselves with a non-parametric method, the path signature [38], that allows for the encoding of path-valued data at almost no loss of information. After introducing and motivating the path signature as a powerful transform for temporal data, we highlight the neglected issue that models employing this transform implicitly interpret discrete time series data as con-

---

[4] However, earlier versions of backpropagation have been around since the 1970s [128].

tinuous paths. We demonstrate this path construction is indeed relevant for learning time series representations that lead to competitive performance in downstream tasks, in particular when working with irregularly spaced time series. Furthermore, we propose a probabilistic approach, using Gaussian processes, that facilitates the creation of beneficial path representations. Albeit this chapter specifically focuses on representation learning with path signatures, its findings can be embedded in the larger paradigm in ML that many models operating on discrete data are actually rooted in (and are approximating) continuous processes and dynamical systems[5].

### 1.2.2 TOPOLOGICAL REPRESENTATION LEARNING

Chapter 6 considers representation learning from a topological perspective. Learned representations are frequently optimised for a downstream classification task, which implies that representations exhibiting high class-separability are sought after. However, in cases where the potentially high-dimensional input data is of interest itself and its structure and shape are yet to be explored, conventionally learned low-dimensional representations (arrived at by means of dimensionality reduction) typically do *not* preserve complex structures in the data space. While linear dimensionality reduction methods (such as principal component analysis (PCA)) may preserve global structures of the input space, they tend to underfit the data manifold due to limited flexibility (as they merely apply affine transformations to the data). Non-linear methods, such as t-distributed stochastic neighbour embedding (t-SNE) or uniform manifold approximation and projection (UMAP), are more flexible and are able to learn local structures in the data, but do so at the cost of losing global structural information. In this chapter, we develop a novel deep learning-based dimensionality reduction method that learns to preserve global structures and shapes of the data space in low-dimensional encodings. This is achieved via a novel topological loss term that can be integrated into end-to-end differentiable autoencoder pipelines. We show that the resulting method, topological autoencoders (TopoAEs), enables the learning of faithful representations that preserve complex structures that can be visually inspected and which are lost in the latent encodings of existing methods.

---

[5] This school of thought has recently brought forth a rich family of continuous analogues of established methods, e.g. Neural ODEs that generalise residual neural networks [37], or Neural CDEs generalising recurrent neural networks [110].

## 1.3 ORGANISATION OF THE THESIS

This thesis is organised in two parts. Part I is dedicated to the classification of clinical time series. It comprises three chapters: Chapter 2 introduces some basic notation, formalises the problem of classifying time series, and introduces the application domain of interest in Part I, the prediction of sepsis. Then, in Chapter 3, as a proof-of-concept study we conduct a single-centre analysis and propose a novel end-to-end differentiable method for the prediction of sepsis that is combining the uncertainty-awareness of Gaussian processes with the inductive bias and efficiency of temporal convolutional networks. Chapter 4 concludes the first part of this thesis with an international, multi-centre study for sepsis prediction using deep learning.

After having focused on clinical predictions (i.e., the ML model's outputs) in Part I, in the subsequent Part II, we consider *internal* representations of ML models. For this, we shift our focus to representation learning, a core aspect of deep learning that is relevant well beyond the deep neural network classifiers that appear throughout Part I.

In Chapter 5, we consider representation learning on time series via *path signatures*, a powerful framework for encoding paths of data. Chapter 6 then takes another perspective on representation learning, namely one from the angle of unsupervised dimensionality reduction. In this chapter, we explore topological methods for learning faithful representations that reveal and preserve shapes and structures in high-dimensional data, that are hard to directly access.

CONTRIBUTIONS This dissertation is partially based on the following publications (ordered by chapter). Additionally, for studies that were included in the thesis, we detail the contributions of individual authors.

- M. Moor$^{\dagger}$, B. Rieck$^{\dagger}$, M. Horn, C. R. Jutzeler$^{\ddagger}$, and K. Borgwardt$^{\ddagger}$. "Early Prediction of Sepsis in the ICU using Machine Learning: A Systematic Review". *Frontiers in Medicine* 8, 2021. DOI: `10.3389/fmed.2021.607952`
  Michael Moor, Bastian Rieck, and Catherine R. Jutzeler performed the data acquisition, extraction, analysis, and interpretation. They drafted the review article. Max Horn substantially contributed to the data interpretation (i.e., quality assessment) and contributed to revising the article. Karsten Borgwardt made significant contributions to the study conception and contributed to the revision of the article. All authors contributed to the writing of the article.

- C. Bock$^{\dagger}$, M. Moor$^{\dagger}$, C. R. Jutzeler, and K. M. Borgwardt. "Machine Learning for Biomedical Time Series Classification: From Shapelets to Deep Learning". In: *Artificial Neural Networks - Third Edition*. Ed. by H. M. Cartwright. Vol. 2190. Methods in

Molecular Biology. Springer, 2021, pp. 33–71. DOI: 10.1007/978-1-0716-0826-5_2
Christian Bock, Michael Moor, Catherine R. Jutzeler and Karsten Borgwardt conceived this book chapter. Christian Bock, Michael Moor, and Catherine R. Jutzeler substantially contributed to the writting of the manuscript draft. All authors contributed to the critical revision and finalisation of the manuscript.

- M. Moor, M. Horn, B. Rieck, D. Roqueiro, and K. Borgwardt. "Early recognition of sepsis with Gaussian process temporal convolutional networks and dynamic time warping". In: *Machine Learning for Healthcare Conference*. PMLR. 2019, pp. 2–26
Michael Moor and Karsten Borgwardt contributed to the conception of the study. Michael Moor and Max Horn contributed to the development and the implementation of the method. Michael Moor preprocessed the data and ran all experiments. All authors substantially contributed to the experimental design, and the interpretation of the empirical results. Bastian Rieck and Damian Roqueiro contributed to the creation of visualisations and illustrations in the manuscript. Michael Moor created the first draft of the manuscript. All authors contributed to the writing, revision, and finalisation of the manuscript.

- M. Moor$^\dagger$, N. Bennett$^\dagger$, D. Plečko$^\dagger$, M. Horn$^\dagger$, B. Rieck, N. Meinshausen, P. Bühlmann, and K. Borgwardt. "Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning". *arXiv preprint arXiv:2107.05230*, 2021
Michael Moor, Nicolas Bennett, Drago Plečko and Karsten Borgwardt conceived the study. Nicolai Meinshausen, Peter Bühlmann, and Karsten Borgwardt supervised the study. Michael Moor, Nicolas Bennett, Drago Plečko, Max Horn, Bastian Rieck, Peter Bühlmann, and Karsten Borgwardt designed the experiments. Nicolas Bennett and Drago Plečko performed the cleaning, harmonisation and label annotation. Michael Moor and Max Horn implemented the filtering and feature extraction. Max Horn and Michael Moor implemented the deep learning models. Michael Moor, Nicolas Bennett, Drago Plečko implemented the baseline ML models. Nicolas Bennett implemented the clinical baselines. Michael Moor designed the patient-focused evaluation. Michael Moor and Bastian Rieck implemented the patient-focused evaluation plots. Bastian Rieck and Michael Moor implemented and designed the performance plots. Bastian Rieck and Max Horn implemented the Shapley value calculation. Bastian Rieck designed, implemented, and performed the Shapley value analysis. Michael Moor ran the internal and external validation experiments for all methods. Michael Moor ran the hyperparameter search of the deep learning models and

LightGBM model. Bastian Rieck ran the hyperparameter search of Logistic regression. Nicolas Bennett investigated different feature sets. Michael Moor implemented and ran the federated (pooling) prediction strategy. Michael Moor designed the pipeline overview figure. Drago Plečko designed the figure regarding unit harmonisation. Bastian Rieck designed the figure regarding the prediction task. Nicolas Bennett designed the risk score illustration and the study flow chart. Drago Plečko devised the dataset table. Peter Bühlmann, and Karsten Borgwardt advised on the algorithmic modelling, statistical interpretation and evaluation. All authors contributed to the interpretation of the findings and to the writing and revision of the manuscript.

- M. Moor, M. Horn, C. Bock, K. Borgwardt, and B. Rieck. "Path Imputation Strategies for Signature Models". In: *ICML Workshop on the Art of Learning with Missing Values*. 2020
  Michael Moor and Bastian Rieck contributed to the conception of this study. Michael Moor implemented the methods and conducted the experiments. Max Horn implemented the code for the datasets. Christian Bock designed the overview figure. All authors contributed to the design of the study and the interpretation of the findings. Michael Moor and Bastian Rieck created the first draft of the manuscript. All authors contributed to the revision and finalisation of the manuscript.

- M. Moor[†], M. Horn[†], B. Rieck[‡], and K. Borgwardt[‡]. "Topological Autoencoders". In: *International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054
  Michael Moor, Max Horn, and Bastian Rieck conceived the study. Michael Moor, Max Horn, and Bastian Rieck designed the method. Max Horn and Michael Moor contributed to the implementation of the different neural network modules. Max Horn and Bastian Rieck implemented the topological loss term. Bastian Rieck (Theorem 1) and Michael Moor (Theorem 2) contributed to the theoretical results. Michael Moor and Bastian Rieck implemented several evaluation metrics. Max Horn and Bastian Rieck designed and implemented the density-based evaluation metric. Michael Moor designed the nested Spheres dataset and implemented the other datasets. Max Horn and Michael Moor implemented and conducted the training and the hyperparameter search. Bastian Rieck substantially contributed to all visualisations in the manuscript. Bastian Rieck provided expertise in topological data analysis. Bastian Rieck and Michael Moor created the first draft of the manuscript. All authors contributed to the writing and to the revision of the manuscript. Bastian Rieck and Karsten Borgwardt supervised the project.

The author of this thesis also contributed to the following publications, that were not prominently reflected in any chapter.

- 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. ISSN: 1367-4803. DOI: `10.1093/bioinformatics/bty246`
- 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*. 2019
- S. L. Hyland†, M. Faltys†, M. Hüser†, X. Lyu†, T. Gumbsch†, C. Esteban, C. Bock, M. Horn, M. Moor, B. Rieck, M. Zimmermann, D. Bodenham, K. Borgwardt‡, G. Rätsch‡, and T. M. Merz‡. "Early prediction of circulatory failure in the intensive care unit using machine learning". *Nature Medicine* 26:3, 2020, pp. 364–373
- M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt. "Set functions for time series". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4353–4363
- 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`
- Z. Wu, Y. Yang, Y. Ma, Y. Liu, R. Zhao, M. Moor, and V. Tresp. "Learning Individualized Treatment Rules with Estimated Translated Inverse Propensity Score". In: *2020 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2020, pp. 1–11
- M. Horn†, E. De Brouwer†, M. Moor, Y. Moreau, B. Rieck†‡, and K. Borgwardt‡. "Topological graph neural networks". *arXiv preprint arXiv:2102.07835*, 2021
- J. Born†, N. Wiedemann†, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, A. Aujayeb, M. Moor, B. Rieck, and K. Borgwardt. "Accelerating detection of lung pathologies with explainable ultrasound image analysis". *Applied Sciences* 11:2, 2021, p. 672
- F. Hensel, M. Moor, and B. Rieck. "A Survey of Topological Machine Learning Methods". *Frontiers in Artificial Intelligence* 4, 2021, p. 52

# Part I

# Clinical time series classification

# 2 PROBLEM FORMULATION

In the first chapter of the part CLINICAL TIME SERIES CLASSIFICATION, we outline two preliminary aspects which will build the foundation of the subsequent sections. We first introduce i) our prediction task: time series classification, and then ii) our application domain: sepsis prediction . This first chapter builds on the content of the following publications:

C. Bock$^{\dagger}$, M. Moor$^{\dagger}$, C. R. Jutzeler, and K. M. Borgwardt. "Machine Learning for Biomedical Time Series Classification: From Shapelets to Deep Learning". In: *Artificial Neural Networks - Third Edition*. Ed. by H. M. Cartwright. Vol. 2190. Methods in Molecular Biology. Springer, 2021, pp. 33–71. DOI: `10.1007/978-1-0716-0826-5\_2`

M. Moor$^{\dagger}$, B. Rieck$^{\dagger}$, M. Horn, C. R. Jutzeler$^{\ddagger}$, and K. Borgwardt$^{\ddagger}$. "Early Prediction of Sepsis in the ICU using Machine Learning: A Systematic Review". *Frontiers in Medicine* 8, 2021. DOI: `10.3389/fmed.2021.607952`

## 2.1 TIME SERIES CLASSIFICATION

### 2.1.1 WHAT IS A TIME SERIES?

Over the last decades, propelled by an ongoing digital revolution, there has been a surge in the collection, curation and distribution of large datasets of biomedical time series. This includes the advent of electronic health record (EHR) databases such as MIMIC [98] or eICU [158], as well as a multiplicity of biosensor data used in remote health monitoring as collected via smartphone apps, wearable sensors or implantable devices [111]. But first off, what is a time series and how is it different from other data?

Time series represent a particular kind of data that typically arise from repeated measurements of a variable of interest. In contrast to sequential data in general, such as DNA sequences in biology or bit strings in computing, measurements in a time series represent data sequences that are equipped with a temporal dimension. Depending on the domain of application, *absolute* measures of time are required, for instance when modelling infection count trajectories in a pandemic wave, or when investigating meteorological time series to forecast the weather. However, there are also domains where *relative* measures of time are more interesting, for instance the relative timing of waves and intervals in an electrocardiogram. In this work, we will focus on the second type of time series, where the temporal component can be interpreted as a relative measure of delay between subsequent measurements.

Adding temporal information to a sequence of data points conveys two main implications: i) the data now has a temporal order (or directionality), and ii) the temporal spacing between subsequent data points introduces a notion of distance between them. At first glance, the first point may strike us as a straight-forward observation. However, when working with time series, being vividly aware of this directionality is crucial. For instance, there are various prediction problems where it is not permissible to utilise data from the future, for instance when trying to predict future movements of stock prices, as this may lead to trivial or circular prediction setups. We henceforth refer to this family of problems as future data leakage. While it may be easy to prevent such a degenerative scenario when using simple and interpretable models for modelling time series, with an increasing complexity of the current state-of-the-art deep learning architectures, detecting future data leakage may not be easy to spot. Regarding the second point, i.e., the temporal spacing between measurements in a time series, this offers an entire battery of challenges an opportunities when working with time series where the temporal spacing is not uniform, i.e., the measurements are not sampled at a fixed time interval like one minute or one hour. Both aspects, temporal ordering and irregular spacing, will resurface throughout the sections of Part I.

### 2.1.2 TIME SERIES NOTATION

Having introduced times series from an eagle's view perspective, we next define time series data in a more formal way in order to clarify basic notation that will reappear throughout this thesis.

**Definition 1** (Time series). *Let* $t \in \mathbb{R}$ *denote a parameter of time, and let* $f(t)\colon \mathbb{R} \to \mathbb{R}^d$ *denote a data generation process with* $d \in \mathbb{N}$ *with* $d \geq 1$. *For a set of discrete times* $\{t_1, \dots, t_l\}$, *we consider* $\mathbf{x}_i = f(t_i)$ *as a measurement value vector observed from* $f$ *at time* $t_i$. *Next, we gather the entries of* $\mathbf{x}_i$ *for* $i \in \{1, \dots l\}$, *such that* $x_{ij}$ *denotes the measurement value of the* $j$*-th dimension at time* $t_i$. *We collect both the observed values and times in vectors* $\mathbf{x} = (x_{11}, \dots, x_{l1}, \dots, x_{1d} \dots, x_{ld})^\top$ *and* $\mathbf{t} = (t_1, \dots, t_1, \dots, t_l \dots t_l)^\top$, *where each of the* $l \cdot d$ *entries of the two vectors represent corresponding inputs and outputs of* $f$. *For* $\mathbf{x}, \mathbf{t} \in \mathbb{R}^{l \cdot d}$, *we then define* $T = (\mathbf{x}, \mathbf{t}) \in \mathcal{T}$ *to be a time series of length* $\operatorname{len}(T) = l$ *and dimension* $\dim(T) = d$.

**Definition 2** (Time series dataset). *Let* $T \in \mathcal{T}$ *be a time series as introduced in Definition 1. A (labeled) time series dataset is then a set* $D = \{(T_1, y_1), (T_2, y_2), \dots, (T_i, y_i), \dots, (T_n, y_n)\}$, *where* $n$ *refers to the number of time series instances, and the pair* $(T_i, y_i)$ *represents the time* *series $T_i$ and its corresponding target $y_i$ of instance $i$. $y_i$ is an element of the target space $\mathcal{Y}$ which is specified by the considered prediction problem and domain.*

**Definition 3** (Subsetted time series)**.** *Let $T \in \mathcal{T}$ be a time series as introduced in Definition 1. By selecting the values* **x** *and times* **t** *only up to an index $p$, where $p$ refers to largest index of* **t** *such that $t_p < k$, we recover a subsetted time series $T^{<k}$ where all times $t_j < k$ for $j \in \{0, 1, \dots, p\}$.*

**Definition 4** (Sparse time series)**.** *Let $T \in \mathcal{T}$ be a time series of length $l$ and dimension $d$ as introduced in Definition 1. For a non-empty set of indices $I$, we discard the corresponding entries from both* **x** *and* **t** *to recover a* sparse *time series $T' \in \mathcal{T}'$ with values and times $\mathbf{x}', \mathbf{t}' \in \mathbb{R}^q$ with $q < l \cdot d$. In contrast to $T$, the value $x'_{ij}$ of $T'$ (corresponding to the $i^{th}$ point in time $t_i$ and the $j^{th}$ dimension) is only* at most *at position $l(j-1)+i$ of $\mathbf{x}'$.*

Using the Definition 1, we refer to a time series as univariate if $d = 1$, and multivariate if $d > 1$. In general, there are several ways how time series are formalised. The utility of a particular choice typically depends on the type of time series that are considered as well as on the specific methods that are applied to them. For instance, if a time series were assumed to have no missing values and consistently shows evenly-spaced time intervals, a matrix notation may be sufficient. When working with time series where not all dimensions are consistently observed, i.e., containing incomplete observations, or additionally exhibiting irregular time intervals between subsequent time steps, definitions that consider time series as a collection of “time, dimension, and measurement”-tuples may be more appropriate. Here, we chose a notation that naturally allows for missing observations (by leaving out an element in both vectors, **x** and **t**) as well as irregular spacing (by incrementing the time values accordingly), but still keeps a vector structure of the observed measurements which will turn out to be useful notation-wise when working with methods such as Gaussian processes in the following chapter.

In Definition 2, we define a time series dataset to be equipped with prediction targets $y_i$, which can represent a class label or a continuous target from some target space $\mathcal{Y}$. In principle, time series datasets could also be considered without target information. However, since we only encounter labeled time series datasets in this thesis, we follow this convention for the sake of notational convenience. Next, for different time series prediction problems, we formulate the task at hand and identify the corresponding input space $\mathcal{T}$ and target space $\mathcal{Y}$, respectively.

### 2.1.3 Time series prediction tasks

Our digitalised times have led us to monitor almost every aspect of modern everyday life. Be it with electronic health records [98], smart watches [164], social networks [71], weather sensors [44], or high-frequency trades [3]; time series data have gained an omnipresence across disciplines and industries. Analysing this data and inferring accurate predictions about targets of interest nowadays can build fortunes and even save lives. However, seemingly similar problems may categorically differ. Therefore, we here introduce and classify a set of frequently employed time series prediction tasks. First, we observe that these tasks can be distinguished by answering the following two questions: i) What exactly is being predicted? ii) Which part of a time series is used as input data for a given prediction?

Whole-series classification To answer the first question, we first consider the most basic setting: *whole series classification* [2]. Here, an entire time series $T \in \mathcal{T}$ is associated with a single class label $y \in \mathcal{Y}$. In the most simple scenario, $\mathcal{Y} = \{0, 1\}$, which means we have binary class labels. However, in many real-world applications, multiclass classification tasks may be encountered with $c$ classes: $\mathcal{Y} = \{0, 1, \ldots, c-1\}$. The goal in whole series classification is to learn a mapping $f \in \mathcal{F}, f\colon \mathcal{T} \to \mathcal{Y}$ such that for a loss function $\ell$, a given time series $T_i$, and a *single* associated class label $y_i$, we aim to find a mapping $f^*$ that minimises the empirical risk $R(f)$:

$$f^* = \operatorname*{arg\,min}_{f \in \mathcal{F}} R(f), \quad \text{where} \tag{2.1}$$

$$R(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(T_i), y_i). \tag{2.2}$$

As a practical example of whole series classification, Boles et al. [16] developed a biometric authentification system based on speech signal.

Window-based classification Next, we introduce a related family of tasks that we refer to as *window-based classification*. Similar to whole series classification, we are provided with a time series $T_i$ and a corresponding, single class label $y_i$. Additionally, the class label has a direct temporal relation to the time series, in that the class label corresponds to an event and is therefore equipped with a time stamp $\tau_i$. Depending on the application, in order to predict an event that occurs at time $t_j$, it may or may not be admissible to employ the entire time series including future data at any time $t_k > t_j$. In medical applications, for instance, it is typically most useful to predict an event using data from the past. Therefore, we consider only time windows up to the event of interest in order to make predictions. For this, we can use the learning objective of Equation 2.1, where we modify the risk calculation to:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(T_i^{<\tau_i}), y_i) \tag{2.3}$$

As a representative example of window-based classification, we refer to a prediction model that monitors patient time series in order to raise an alarm if a complication is inbound [142]. Here, during training, a model was provided with a subsetted time series up until the event of interest, as well as the event itself, i.e., the time-stamped class label.

PER-TIMEPOINT CLASSIFICATION As a third type of time series classification problem, we consider per-timepoint classification. Here, each available time step is associated with a prediction target $y_{it}$, which we collect per instance in a vector $\mathbf{y}_i$. The goal is then to solve Equation 2.1, where we the risk calculation needs to account for the per-timepoint predictions:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\sum_{k=1}^{m_i} \ell\left(f\left(T_i^{<t_k}\right), y_{ik}\right), \tag{2.4}$$

where we emphasise that different instances may have varying-length time series by setting $m_i = \mathrm{len}(T_i)$. To give an example of a per-timepoint classification setting, clinical early warning systems have been trained to either raise or not raise an alarm at each point of a patient time series [95]. Finally, we visually highlight the differences between these three types of time series classification tasks in Figure 2.1.

FURTHER TIME SERIES TASKS Since this chapter aims to give an introduction to time series classification, here we collect and only briefly mention an non-exhaustive list of other types of prediction tasks that may be encountered when analysing time series. In time series forecasting, the goal is to predict future time series values based on the present and past ones. Here, the goal is to learn a mapping $f\colon \mathcal{T} \to \mathcal{T}$. Forecasting has been widely used in financial time series analyses such as stock market predictions [163], as well as for weather forecasting [25]. As another task, in time series dimensionality reduction, the goal is to find a latent representation of the time series that captures the relevant signal while discarding noise and redundant information. This may be intended for reducing the dimensions (and noise) along the time axis [105], or along the time series channel dimensions in multivariate time series [150]. In time series regression, the goal is to predict a continuous target, for instance $y \in \mathbb{R}$, based on an input time series $T$. In the most simple case, we learn a

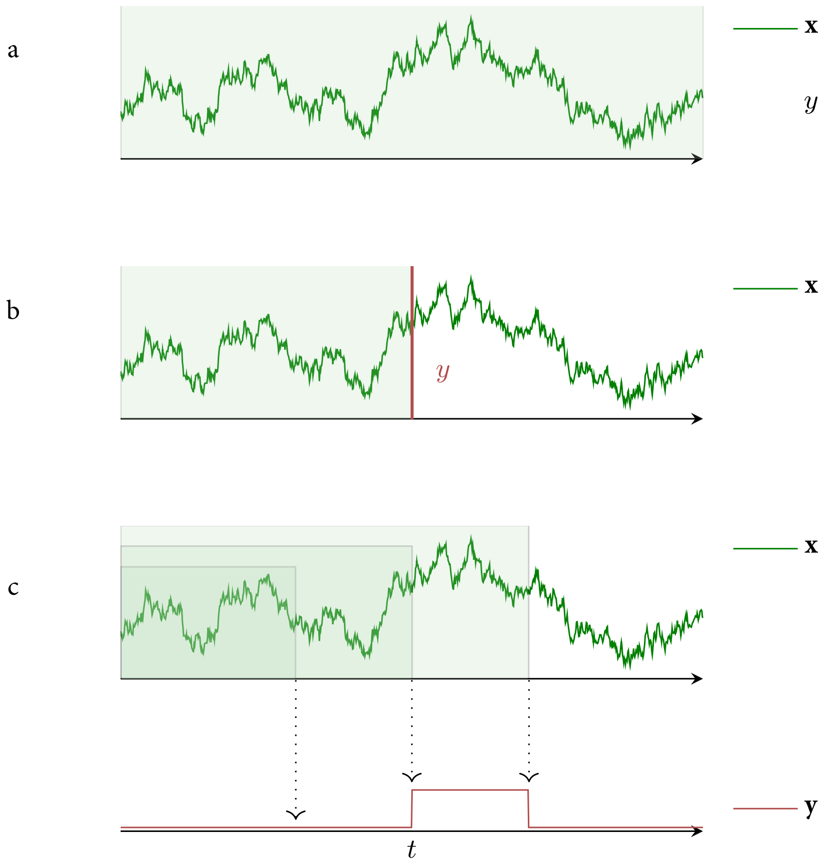


Figure 2.1: Illustration of three types of time series classification tasks. Time-varying variables are written as vectors in bold. Panel a: Whole series classification. The time series values $\mathbf{x}$ are provided with a label $y$. Panel b: Window-based classification. The time series values $\mathbf{x}$ are provided with a time-stamped label $y$ (red). Panel c: Per-timepoint classification. Both the input time series $\mathbf{x}$ and the labels $\mathbf{y}$ are sequences. Three predictions are highlighted where all data up until the given point in time is used for prediction. The filled boxes (green) indicate which windows of time series data are used for prediction.

mapping $f : \mathcal{T} \rightarrow \mathbb{R}$. Assuming continuous time series values (as opposed to discrete ones), other tasks such as forecasting and dimensionality reduction may also be subsumed as special cases of regression, however with varied specifications of the target space $\mathcal{Y}$. From a mathematical viewpoint, regression may even be subsumed as a special case of multiclass classification where the number of classes $c \rightarrow \inf$. However, in practice the categorical distinction is well justified. For instance, while classifiers may have an internal representation of each class (as a neuron in the output layer of neural network), this may not be feasible when $c \rightarrow \inf$.

### 2.1.4 METHODS FOR TIME SERIES CLASSIFICATION

After having familiarised ourselves with different types of time series prediction problems, and this with a particular focus on time series classification, we next give an introduction to methods that are used to solve time series classification tasks. This involves both data mining and classical machine learning techniques as well as deep learning methods. Time series classification algorithms can be subdivided into *feature-based* and *distance-based* approaches [14].

FEATURE-BASED METHODS Feature-based methods first extract temporal statistical properties of the time series to collect them in feature vectors, such that a time series, or an extracted time window thereof, can be represented as an element of a vector space. Subsequently, a generic classifier may be trained directly on the feature representations, potentially oblivious to the sequential nature of the raw time series data. Examples of such feature mappings include basic summary statistics like mean, median, maximum, minimum, and variance of the time series over multiple look-back windows [95, 138]. Furthermore, more involved feature mappings have been employed, that are based on the discrete Fourier transform [212], the wavelet transform [116, 144], path signatures [38], or topological data analysis (TDA) [133], to name a few.

DISTANCE-BASED METHODS Conventionally, feature-based methods have been contrasted by distance-based methods, where instead of an explicit feature mapping, distances or dissimilarities between time series are used to classify them. As a famous and powerful example of such a distance measure, in dynamic time warping (DTW), two time series are aligned using dynamic programming [11], where the "cost" of alignment determines the resulting distance between two time series. Combining DTW with a $k$-nearest neighbour classifier (DTW-$k$NN) turned out to become a strong baseline for time series classification ever since [106]. Gudmundsson et al. [73] investigated combining DTW with kernel-based classifiers such as the support vector machine (SVM), however as DTW is not a *metric*, including it in kernel methods requires some additional care. In more recent works, optimal transport has been used to derive distances by measuring the cost of transforming one time series into another one, for instance in terms of matching the distributions of their subsequences, as proposed with the Wasserstein time series kernel (WTK) [14]. Also here, the distance measure does not fulfill the criteria of a metric, potentially resulting in indefinite kernel matrices, which has been addressed via the usage of Kreĭn SVMs [149].

This categorisation of time series classification methods into feature-based and distance-based can be criticised for being an oversimplification. For instance, a $k$-nearest neighbor ($k$NN) classifier can be applied to feature representations of time series in order to predict the class label of a time series in terms of its nearest neighbours with respect to some distance measure. Conversely, distance measures may also be used to construct feature representations. The fuzziness of this distinction becomes even more evident when considering deep learning techniques.

DEEP LEARNING METHODS The advent of deep learning methods has drastically changed the playing field for time series classification. As one striking advantage, neural networks allow for an automatic and learnable feature extraction step. So they can be seen as feature-based methods, where the feature mapping itself is also learned. Various types of deep neural network architectures have been optimised for sequential data, and have found wide adoption in time series classification. For example, this includes recurrent neural networks (RNNs) such as long short-term memory networks (LSTMs) [86], temporal convolutional networks (TCNs) [7], and attention models [201]. Interestingly, also for deep learning methods distance-based learning can play a role, for instance when performing contrastive learning [107], where pairs of similar or dissimilar instances are sought to be encoded similarly or dissimilarly, respectively.

## 2.2 EARLY PREDICTION OF SEPSIS

Part I of this thesis is concerned with clinical time series classification. As the first part of this chapter has more generally outlined time series classification tasks and methods, in the following sections, we introduce our medical application case, sepsis. We then formulate our clinical prediction problem of interest and give an overview to existing approaches.

### 2.2.1 What is sepsis?

History of sepsis Infectious diseases have been a leading cause of mortality by rampaging through communities world-wide throughout the history of human civilisation. Sepsis refers to a potentially fatal complication of a severe, typically bacterial infection, and was most recently defined to be a dysregulated host response to infection [185]. Even *millenia* before the Black Death Plague has wiped out roughly a third of the population of medieval Europe and Asia, ancient scholars have already been aware and afraid of sepsis [60]. The word sepsis itself stems from σηψις which is Greek for the "decomposition of animal or vegetable organic matter in the presence of bacteria." [68]. The word sepsis was first encountered in a medical context in Homer's poems approximately 2,700 years ago where it appears as derived from the verb *sepo* (σηπω) which translates to "I rot" [68]. Even though sepsis kept a presence over the centuries, for instance by appearing in the writings of Hippocrates and Galen, only with the golden age of germ theory in the 1800s its microbial origins were starting to get unravelled [60].

Defining sepsis: from Sepsis-1 to Sepsis-3 While the ancient depiction of sepsis was mostly characterised by decaying flesh, wounds, and fever, the advent of modern microbiology and immunology allowed researchers to draw a refined picture of sepsis. First, this involved the clarification that contagious bacterial infections were the foundation of a septic complication. Then, during the 20th century, scholars gained deeper insights into the molecular processes of sepsis uncovering the roles of cytokines and the coagulation system [60]. In the early 1990s, the first international consensus definition of sepsis, Sepsis-1, described sepsis as an infection-induced systemic inflammation [17]. Due to systemic inflammation being not specific enough for sepsis, i.e., by also presenting in patients without infection that have a better prognosis, Sepsis-2 was proposed to extend the first definition by considering further inflammatory, hemodynamic, and organ dysfunction parameters [122]. The newly proposed term for sepsis aimed to increase specificity by including indicators of organ injury. Nevertheless, the previous criteria for defining sepsis were kept in use, which led to confusions, for instance whether to diagnose "sepsis" by the new diagnostic criteria, or "severe sepsis" by the old ones [74]. A long-awaited reconsideration of sepsis finally occurred with the Sepsis-3 definition in 2016 [185], where sepsis was defined as a life-threatening, dysregulated host response to infection. This was operationalised by the evaluation of the sequential organ failure assessment (SOFA) [204], where the clinical diagnostic criteria of sepsis comprise an acute increase of at least two points in SOFA combined with a suspected infection [185]. Throughout this thesis, we will consider the most recent definition, Sepsis-3. That being said, also this most recent definition has its limitations. Compared to previous definitions, Sepsis-3 by design implies organ dysfunction, making it a more narrow definition describing a more severe cohort. Next, as a relevant limitation to Sepsis-2 and Sepsis-3, the requirement of a suspected (or proven) infection introduces a strong dependence of these definitions on the clinicians diagnostic interventions (e.g. blood culture sampling). Finally, the complexity and multi-modality of Sepsis-3, which employs the SOFA score, renders this definition most applicable in the intensive care unit (ICU). However, defining sepsis in settings with scarcer data remains challenging.

Sepsis: a persistent dilemma Even though sepsis has worn many faces over the past centuries, one essential property has been preserved. In fact, one which was already known to Niccolo Machiavelli, a political writer during the 16th century. In a famous quote of his contemporary physicians he stated: "As the physicians say of hectic fever, that in the beginning of the malady it is difficult to detect but easy to treat, but in the course of time, having been neither detected nor treated in the beginning, it becomes easy to detect but difficult to treat." [186]. Despite the groundbreaking advances of modern medicine such as antimicrobial and immunomodulatory therapies or intensive care monitoring, clinicians today are still facing the same crux. In the hard-to-identify early stages of sepsis, organ damage may still be reversible such that an effective antimicrobial therapy leads to improved outcomes. However, each hour of delayed intervention leads to a measurable increase in mortality [54].

### 2.2.2 Related work on the early prediction of sepsis

Over the last decades, clinicians and researchers have been searching for biomarkers that would allow for an early recognition of sepsis, albeit with little success [21, 200]. Now, amidst an ongoing digital revolution in healthcare, and driven by the routine collection of patient data in patient data management systems and electronic health record databases, there is a new hope for tackling the problem of early recognition by uncovering *digital* biomarkers of sepsis. For this, the idea is to mine the plethora of streaming patient data using machine learning (ML) techniques in order to leverage early signals that are predictive of an imminent sepsis.

Early studies employing ML in sepsis patients focused on predicting sepsis-related clinical outcomes such as mortality [75, 168]. Even though such a hard endpoint as mortality is clearly defined and a natural choice for a clinical prediction target, one may wonder to which degree the accurate prediction of in-hospital mortality in sepsis patients is actually clinically actionable in that it supports the decision-making of clinicians. In particular since sepsis management is a highly time sensitive situation, where each hour of delayed antibiotic treatment and fluid resuscitation can lead to potentially irreversible organ damage, there is a strong practical argument in favour of predicting if and when exactly sepsis will occur, rather than foreseeing its terminal outcome.

Propelled by a previously unseen availability of large EHR databases such as MIMIC [70, 98], initial studies were conducted that considered sepsis prediction as an early warning problem. Henry et al. [82] developed a targeted real-time warning score (TREWScore) for predicting septic shock with a cox proportional hazard model. Next, a risk model named "InSight" was developed to predict future onsets of sepsis [24]. These initial studies still relied on conventional statistical modelling techniques, and Calvert et al. [24] considered only a small set of nine vital signs. In the subsequent years, more large-scale approaches using deep learning would gain considerable attention [61, 101]. At this point in time and at this stage of the literature, my doctoral studies have begun. Following a chronological order, in the next chapter, we elucidate the first sepsis prediction project of my doctoral studies.

# 3 UNCERTAINTY-AWARE RECOGNITION OF SEPSIS WITH GAUSSIAN PROCESS TEMPORAL CONVOLUTIONAL NETWORKS

In this chapter, we present Gaussian process temporal convolutional networks (MGP-TCNs) [142], a deep learning approach for the early prediction of sepsis. The content of this chapter is based on the following publication:
M. Moor, M. Horn, B. Rieck, D. Roqueiro, and K. Borgwardt. “Early recognition of sepsis with Gaussian process temporal convolutional networks and dynamic time warping”. In: *Machine Learning for Healthcare Conference*. PMLR. 2019, pp. 2–26

After having introduced sepsis in Section 2.2, this chapter provides a more in-depth treatment of the prediction task at hand, as well as the gaps in the previous literature before elucidating the proposed method and the conducted experiments. Due to an immense affluence of data, the projects of this thesis evolving around the application of sepsis prediction, that is, the chapters of Part I, are focusing on data from the intensive care unit (ICU). The remaining chapter is organised as follows: In Section 3.1, we give a brief introduction to our prediction problem and the related literature. Then, in Sections 3.2 and 3.3, we familiarise ourselves with Gaussian processes and temporal convolutional networks, respectively. Next, in Section 3.4 we introduce the method MGP-TCN and in Section 3.5, we outline the experimental setup and present and discuss the empirical results. Section 3.6 discusses the results and concludes the chapter with final remarks.

## 3.1 INTRODUCTION

Despite decades of research, sepsis remains a public health crisis with significant mortality, morbidity and associated health costs [47, 94, 103]. There is mounting evidence that effective sepsis management requires an early diagnosis followed by a rapid initiation of an effective antimicrobial therapy [167]. However, recognising sepsis patients in the early stages, where organ damage is still reversible, is a notoriously difficult task for clinicians. While early signs and symptoms may still be vague and unspecific, and the clinician is still absorbed in a broad differential diagnosis, too often there is a delay in the potentially life-saving initiation of antimicrobial and resuscitation therapy [134].

Given the above disposition and the urgent clinical need for early and accurate warnings, sepsis prediction has gained attention from the machine learning community [61, 101]. For this, the task of predicting sepsis was commonly formulated as a multi-channel time series classification task based on vital, demographic, and laboratory patient data. Due to irregularly observed and noisy time series data, a set of hand-crafted preprocessing steps including the resampling of time series into bins of fixed size, carry-forward imputation of missing values, and smoothing of noisy time series via rolling means have been employed [24, 48]. However, the process of how one arrives at the exact preprocessing strategy and corresponding hyperparameters is typically not evident. Furthermore, imputing missing values could lead to the loss of valuable information about informative data missingness. For instance, the presence or absence of certain measurements could be associated with the patient state. Futoma et al. [61] proposed the first sepsis prediction model that accounted for irregular sampling using Gaussian process adapters. This framework enabled the imputation of missing values while preserving uncertainty (due to missingness) in the down-stream classifier, a LSTM model, which was then used to predict sepsis.

Variants of recurrent neural networks (RNNs) have been exerting dominance in various time series and sequence modelling tasks over the previous decades [86]. However, convolutional neural networks have recently gained attention for these tasks. In particular, temporal convolutional networks (TCNs) [117], a class of convolutional models, have been shown to be able to outperform conventional recurrent neural networks (RNNs) in several sequential tasks in terms of performance metrics, memory efficiency, and parallelism [7][1].

Building on these developments, in this chapter we propose MGP-TCN, an end-to-end trainable deep learning model for sepsis prediction on irregularly-sampled, multivariate time series, that combines the uncertainty awareness of multi-task Gaussian process (MGP) adapters with TCNs. In addition, we present a lazy learner multi-channel ensemble based on dynamic time warping $k$-nearest neighbor (DTW-$k$NN), an established data mining technique. Furthermore, we propose the first fully-accessible benchmark for sepsis early detection by making temporally resolved Sepsis-3 labels publicly available[2]. Finally, in a thorough experimental evaluation, we empirically demonstrate how the proposed approaches outperform the hitherto state-of-the-art method for sepsis prediction. Before

---

[1] It is worth mentioning at this point, that in the mean time, inspired by natural language processing, attention models [189, 201] have become a go-to solution for many sequential tasks (see Chapter 4).

[2] See `https://github.com/BorgwardtLab/mgp-tcn`

diving into the methods and experiments, we pave the way by giving some background on Gaussian processes and TCNs.

## 3.2 GAUSSIAN PROCESSES

Gaussian processes represent a powerful and flexible class of statistical models that over the last decades have found wide adoption throughout technical disciplines and the natural sciences. In this section, we first give an introduction to Gaussian process (GP) regression as based on the book by Rasmussen et al. [162]. Then, we consider their extension to multi-task learning [209], and conclude with GP adapters [125].

### 3.2.1 GAUSSIAN PROCESS REGRESSION

**Definition 5.** *A Gaussian process is a collection of random variables, any finite subset of which is jointly Gaussian.*

Consequently, a Gaussian process is fully determined by its mean function $m(t)$ and covariance function $k(t, t')$, so that we can write:

$$f(t) \sim \mathcal{GP}\big(m(t), k\big(t, t'\big)\big). \tag{3.1}$$

Here, the random variables represent the values of a function $f$ evaluated at locations $t$. For our intended purposes, the index set of these random variables can be interpreted as time, which is why we use $t$ to denote the location. In general, however, the location or input space of a GP may take other forms, and could for example be $\mathbb{R}^d$. Equation 3.1 illustrates that a GP can be interpreted to be modelling a distribution over *functions*. For notational and computational convenience, the mean function is often assumed to be zero: $m(t) = 0$. Then, the specification of a covariance function allows us to draw sample functions for a given vector of $r$ queried locations $\mathbf{t}_*$ with

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{t}_*\mathbf{t}_*}), \tag{3.2}$$

where $\mathbf{f}_* = (f(t_{*_1}), \ldots, f(t_{*_r}))^\top$ collects the function evaluations at the times $\mathbf{t}_*$ and $\mathbf{K}_{\mathbf{t}_*\mathbf{t}_*}$ refers to the $r \times r$ covariance matrix evaluated at the query locations $\mathbf{t}_*$ using the covariance function $k(\cdot, \cdot)$. As those function draws carry little information of interest, they are referred to as samples from the GP "prior". It becomes more interesting when extending our set of query times with training data $(\mathbf{x}, \mathbf{t})$, where $\mathbf{x}$ refers to observed values, and $\mathbf{t}$ refers to the corresponding locations in time (we follow the notation introduced in Section 2.1.2, Definition 1). Assuming a noise-free scenario with $x_i = f(t_i) + \epsilon$ with $\epsilon \to 0$, we have $\mathbf{x} = \mathbf{f}$, which means we can model the following joint distribution

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{tt}} & \mathbf{K}_{\mathbf{tt}_*} \\ \mathbf{K}_{\mathbf{t}_*\mathbf{t}} & \mathbf{K}_{\mathbf{t}_*\mathbf{t}_*} \end{bmatrix}\right), \tag{3.3}$$

where $\mathbf{K}_{\mathbf{tt}_*}$ refers to the $n \times r$ covariance matrix evaluated between the $n$ training locations $\mathbf{t}$ and the $r$ query points $\mathbf{t}_*$, and so on. Making use of Gaussian identities (see Rasmussen et al. [162, Section A.2]), we can derive the conditional distribution $\mathbf{f}_*|\mathbf{t}_*, \mathbf{t}, \mathbf{f}$ that represents the distribution of function values at query times $\mathbf{t}_*$ when conditioning on the information of the observed data $(\mathbf{x}, \mathbf{t})$. This distribution is therefore referred to as the "posterior" and can be written as:

$$\mathbf{f}_*|\mathbf{t}_*, \mathbf{t}, \mathbf{f} \sim \mathcal{N}\big(\mathbf{K}_{\mathbf{t}_*\mathbf{t}}\mathbf{K}_{\mathbf{tt}}^{-1}\mathbf{f},\ \mathbf{K}_{\mathbf{t}_*\mathbf{t}_*} - \mathbf{K}_{\mathbf{t}_*\mathbf{t}}\mathbf{K}_{\mathbf{tt}}^{-1}\mathbf{K}_{\mathbf{tt}_*}\big) \tag{3.4}$$

In a more realistic scenario, we consider noisy observations using additive independent identically distributed Gaussian noise $\epsilon$ with variance $\sigma^2$. This leads to the following modification in the covariance of the observed values

$$\mathrm{cov}(\mathbf{x}) = \mathbf{K}_{\mathbf{tt}} + \sigma^2\mathbf{I}, \tag{3.5}$$

where $\mathbf{I}$ represents the $n \times n$ identity matrix. Accordingly, the predictive distribution then evaluates to:

$$\mathbf{f}_*|\mathbf{t}_*, \mathbf{t}, \mathbf{x} \sim \mathcal{N}\Big(\mathbf{K}_{\mathbf{t}_*\mathbf{t}}\big(\mathbf{K}_{\mathbf{tt}} + \sigma^2\mathbf{I}\big)^{-1}\mathbf{x},\ \mathbf{K}_{\mathbf{t}_*\mathbf{t}_*} - \mathbf{K}_{\mathbf{t}_*\mathbf{t}}\big(\mathbf{K}_{\mathbf{tt}} + \sigma^2\mathbf{I}\big)^{-1}\mathbf{K}_{\mathbf{tt}_*}\Big) \tag{3.6}$$

While GPs regression models are lazy learners that produce predictions directly as a function of the training data and the queried locations, the covariance function still has hyperparameters that may be learnt. For instance, this may include a length scale, as well as signal and noise variances. Conventionally, hyperparameters of GPs are tuned by maximising the marginal likelihood $p(\mathbf{x}|\mathbf{t})$, the model's probability of the observed values given the locations and marginalised over the functions $\mathbf{f}$. The marginal likelihood is an integral of the likelihood (of the observed data given the function $\mathbf{f}$) times the prior of $\mathbf{f}$:

$$p(\mathbf{x}|\mathbf{t}) = \int p(\mathbf{x}|\mathbf{f}, \mathbf{t})p(\mathbf{f}|\mathbf{t})\,\mathrm{d}\mathbf{f}. \tag{3.7}$$

Conveniently, the log marginal likelihood can be expressed in closed form (see Rasmussen et al. [162, Section 2.2]):

$$\log p(\mathbf{x} \mid \mathbf{t}) = -\frac{1}{2}\mathbf{x}^\top \left(\mathbf{K}_{\mathbf{tt}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{x} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{tt}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi \tag{3.8}$$

Upon implementation, the matrix inversion is commonly replaced by a Cholesky decomposition for numerical stability [162]. Nevertheless, in this so-called *exact inference* setting, we can expect a runtime complexity of $\mathcal{O}\left(n^3\right)$. The cubic runtime can quickly become excessive in real-world data. Therefore, over the last decades, various approximative inference strategies have been proposed to reduce this cost. For instance, inducing point methods [160] use only a small number of $m < n$ inducing points to learn a rank $m$ approximation of the covariance matrix leading to $\mathcal{O}\left(nm^2\right)$ [187]. Furthermore, structured kernel interpolation even leads to $\mathcal{O}(n + m \log m)$ for a grid of $m$ inducing points [210].

While GPs can be used to perform regression (or classification) tasks in general, in both Part I and Part II of this thesis, we are mainly interested in using GPs to model the data generating process which then can be used to impute missing values of irregularly-sampled time series while preserving the model's Bayesian uncertainty at each imputed point. In our applications, we are generally interested in multivariate time series of dimension $d > 1$. Therefore, we next consider a useful extension to GPs that allows for modelling high-dimensional processes.

### 3.2.2 MULTI-TASK GAUSSIAN PROCESSES

Multi-task Gaussian processes (MGPs) were proposed as an extension of GP inference to multi-task learning [18]. To build intuition from the start, in the following description, *tasks* can be interpreted to correspond to channels of a multivariate data generating process of which we only observe discrete measurements as a multivariate time series.

Given $n$ distinct inputs $t_1, \ldots, t_n$, we gather the complete set of responses (or time series values) for $m$ tasks (or channels) in a vector $\mathbf{x} = (x_{11}, \ldots, x_{n1}, \ldots, x_{1m}, \ldots, x_{nm})$. Here, $\mathbf{x}$ can be interpreted as the vector of observed values, consistent with Definition 1. We place a GP prior over $m$ latent functions $\{f_1, \ldots, f_m\}$ in order to induce correlations between the different tasks. Assuming a mean function of zero, we have

$$\operatorname{cov}\left(f_p(t), f_q\left(t'\right)\right) = \left(\mathbf{K}^f\right)_{pq} \cdot k\left(t, t'\right), \tag{3.9}$$

where $\mathbf{K}^f$ is a $m \times m$ positive semi-definite matrix representing the similarities between the tasks and therefore between the latent functions, and $k(\cdot, \cdot)$ represents a covariance function over the inputs (or time locations). Further, we model $x_{ip}$, the $i^{\text{th}}$ observation of task $p$, as

$$x_{ip} \sim \mathcal{N}\big(f_p(t_i),\ \sigma_p^2\big), \tag{3.10}$$

where $\sigma_p^2$ indicates the noise variance of task $p$. To conform with the single-task GP formulation in the preceding Section 3.2.1, we collect the covariances over the inputs in matrices as follows and indicate the used location vectors:

$$(\mathbf{K}_{\mathbf{tt}'})_{ij} = k\big(t_i, t'_j\big). \tag{3.11}$$

Then, in the fully-observed case, the responses $\mathbf{x}$ at locations $\mathbf{t}$ follow

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0},\ \mathbf{\Sigma}), \tag{3.12}$$

with

$$\mathbf{\Sigma} = \mathbf{K}^f \otimes \mathbf{K}_{\mathbf{tt}} + \mathbf{D} \otimes \mathbf{I}, \tag{3.13}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{K}_{\mathbf{tt}}$ is the $n \times n$ matrix of covariances between all training locations, and $\mathbf{D}$ represents an $m \times m$ diagonal matrix with the $(p, p)^{\text{th}}$ entry being $\sigma_p^2$. Due to the Kronecker product, $\mathbf{\Sigma}$ is an $mn \times mn$ covariance matrix, from which we can already see that its naive inversion will cost $\mathcal{O}\big(m^3 n^3\big)$.

Regarding the predictive distribution, for notational convenience, we use the abbreviation $\mathbf{z} := \mathbf{f}_* | \mathbf{t}_*, \mathbf{t}, \mathbf{x}$. Then, for queried points $\mathbf{t}_*$ the predictive distribution becomes

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}},\ \mathbf{\Sigma}_{\mathbf{z}}), \tag{3.14}$$

where

$$\boldsymbol{\mu}_{\mathbf{z}} = \Big(\mathbf{K}^f \otimes \mathbf{K}_{\mathbf{t}_*\mathbf{t}}\Big)\mathbf{\Sigma}^{-1}\mathbf{x}, \tag{3.15}$$

and

$$\mathbf{\Sigma}_{\mathbf{z}} = \Big(\mathbf{K}^f \otimes \mathbf{K}_{\mathbf{t}_*\mathbf{t}_*}\Big) - \Big(\mathbf{K}^f \otimes \mathbf{K}_{\mathbf{t}_*\mathbf{t}}\Big)\mathbf{\Sigma}^{-1}\Big(\mathbf{K}^f \otimes \mathbf{K}_{\mathbf{t}\mathbf{t}_*}\Big). \tag{3.16}$$

Bonilla et al. [18] showed that also for MGPs, a closed-form expression for the marginal likelihood can be derived. Interestingly, the noise in MGPs is *crucial* for enabling the sharing of information across tasks: When considering noise-free observations in a block design (i.e., considering the same locations for all $m$ tasks), it has been shown that upon maximising marginal likelihood, there is no transfer between the tasks [18], leaving us with $m$ independent single-task GPs. This phenomenon is also known as *autokrigeability* in the geostatistics literature [205].

### 3.2.3 GAUSSIAN PROCESSES ADAPTERS

Building on the previous sections on Gaussian processes, we now consider Gaussian process adapters, a framework proposed by Li et al. [125] for the classification of irregularly-sampled and sparse time series. While Li et al. [125] introduced GP adapters using single-task GPs, we directly introduce their multi-task extension [61]. To start with an informal summary, in a first step, GP regression is used to derive a predictive distribution over evenly-spaced imputations of an irregularly-sampled time series. These imputed time series are then used for time series classification. Here, the GP hyperparameters are optimised end-to-end using the downstream classification task [125]. Having roughly sketched the main idea, we now give a formal introduction to GP adapters.

Following Definitions 1 and 2, let $D = \{(T_1, y_1), \dots, (T_n, y_n)\}$ be a dataset of time series. We assume the time series to be of varying length, that is, $\text{len}(T_i) \neq \text{len}(T_j)$ for some $i, j \in \{1, \dots, n\}$. Further, we assume that the time series $T_i$ is irregularly spaced, i.e., there are time indices $j$ in $\mathbf{t}_i$ for which $t_j - t_{j-1} \neq t_{j+1} - t_j$. Note that since we allow for multivariate time series, the time vector $\mathbf{t}_i$ can have repeated time values such that at the $k^{\text{th}}$ position of the time vector $(\mathbf{t}_i)_k = t_{k'}$, the corresponding time index $k'$ generally satisfies $k' \leq k$. Thus, the above triplet refers to subsequent time indices and *not* positional indices in the time vector $\mathbf{t}_i$. Next, we consider instance $i$ and for notational convenience omit the instance index, e.g. $T := T_i$. Then, we fix a set of $r$ evenly-spaced reference locations $\mathbf{t}_* = (t_{*_1}, \dots, t_{*_r})^\top$. The goal is then to represent our $d$-dimensional time series $T$ as the MGP posterior distribution of $d$ tasks queried at those $r$ reference locations. For this, we employ a zero-mean MGP prior and use a covariance function $k(\cdot, \cdot)$. Using Equations 3.9 to 3.16, the MGP allows us to model a latent time series $T_* = (\mathbf{z}, \mathbf{t}_*)$, where $\mathbf{z}$ refers to the imputed time series values that follow the MGP posterior distribution as shown in Equations 3.14 to 3.16. While the covariance matrices over the inputs are computed for each instance $i$ individually, the task similarity matrix $\mathbf{K}^f$, the task-specific noise variances $\{\sigma_1^2, \dots, \sigma_d^2\}$, as well as the parameters $\boldsymbol{\eta}$ of the covariance function $k(\cdot, \cdot)$ are shared across all instances and treated as the MGP hyperparameters $\boldsymbol{\theta}$ to be learned

$$\boldsymbol{\theta} = \text{vec}\left(\mathbf{K}^f\right) \oplus \left(\sigma_1^2, \dots, \sigma_d^2\right)^\top \oplus \boldsymbol{\eta}, \tag{3.17}$$

where $\oplus$ denotes the vector concatenation operation and $\text{vec}(\cdot)$ denotes the vectorisation of a matrix. This rather general parametrisation of the MGP may in practice be simplified (in terms of the number of free parameters), for instance by using a Cholesky decomposition $\mathbf{K}^f = \mathbf{L}\mathbf{L}^\top$ where $\mathbf{L}$ is lower triangular [18].

Next, we address how the GP posterior $\mathbf{z}$ is used to actually impute time series data at evenly-spaced times in order to classify the time series while preserving uncertainty about data missingness. We wish to learn a mapping $f\colon \mathcal{T}' \to \mathcal{Y}$ that allows us to classify an irregularly-observed time series $T' \in \mathcal{T}'$. GP adapters address this problem via the composition of two mappings $g_{\boldsymbol{\theta}}\colon \mathcal{T}' \to \mathcal{T}$ and $h_{\boldsymbol{\phi}}\colon \mathcal{T} \to \mathcal{Y}$ such that $f = h_{\boldsymbol{\phi}} \circ g_{\boldsymbol{\theta}}$ where $\boldsymbol{\theta}, \boldsymbol{\phi}$ refer to the parameters, respectively. Here, $g_{\boldsymbol{\theta}}\colon (\mathbf{x}, \mathbf{t}) \mapsto (\mathbf{z}, \mathbf{t}_*)$, i.e., given a irregular input time series $T' = (\mathbf{x}, \mathbf{t})$[3], $g_{\boldsymbol{\theta}}$ returns regular grid of query times $\mathbf{t}_*$ and a corresponding vector of imputed time series values $\mathbf{z}$ which is drawn following the MGP posterior distribution[4]. $h_{\boldsymbol{\phi}}$ may be realised with any black-box classifier that can leverage evenly-spaced multivariate time series to output a predicted class label. For the sake of notational convenience, given instance $i$, we assume that $h_{\boldsymbol{\phi}}$ can directly use the "flattened" vector $\mathbf{z}_i$, that is, it internally applies the reshaping operation, in case a $r \times d$ matrix format is required. Were $\mathbf{z}_i$ directly observed, given a loss function $\ell$, we could directly apply the classifier $h_{\boldsymbol{\phi}}$ to evaluate $\ell(h_{\boldsymbol{\phi}}(\mathbf{z}_i), y_i)$. However, in our case $\mathbf{z}_i$ is a Gaussian random vector, making the loss $\ell$ also a random variable given a target $y_i$. This is accounted for by using the expectation $\mathbb{E}_{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}_i}, \boldsymbol{\Sigma}_{\mathbf{z}_i}; \boldsymbol{\theta})}[\ell(h_{\boldsymbol{\phi}}(\mathbf{z}_i), y_i)]$ as the overall loss for optimisation. The learning problem then becomes

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}} = \operatorname*{arg\,min}_{\boldsymbol{\phi}, \boldsymbol{\theta}} \sum_{i=1}^{n} \overbrace{\mathbb{E}_{\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}_i}, \boldsymbol{\Sigma}_{\mathbf{z}_i}; \boldsymbol{\theta})} \left[\ell(h_{\boldsymbol{\phi}}(\mathbf{z}_i), y_i)\right]}^{E_i}, \tag{3.18}$$

where both the parameters $\boldsymbol{\theta}$ of the MGP imputation $g_{\boldsymbol{\theta}}$ as well as the parameters $\boldsymbol{\phi}$ of the classifier $h_{\boldsymbol{\phi}}$ are optimised *jointly*. Notably, for many choices of $h_{\boldsymbol{\phi}}$ the above expectation (abbreviated with $E_i$) is analytically not tractable [61, 125]. Therefore, this term is approximated with Monte Carlo sampling using $s_m$ samples

$$E_i \approx \frac{1}{s_m} \sum_{s=1}^{s_m} \ell\Big(h_{\boldsymbol{\phi}}\Big(\mathbf{z}_i^{(s)}\Big), y_i\Big), \tag{3.19}$$

---

[3] For notational convenience, irregular sampling is not explicitly indicated for $\mathbf{x}$ and $\mathbf{t}$.

[4] The query locations can actually be considered as an argument of $g_{\boldsymbol{\theta}}$, since the MGP conditions its posterior on the queried locations $\mathbf{t}_*$. However, for notational convenience we define $g_{\boldsymbol{\theta}}$ to simply map from the space of irregular time series to the space of regular (evenly-spaced) time series.

where each

$$\mathbf{z}_i^{(s)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{z}_i}, \boldsymbol{\Sigma}_{\mathbf{z}_i}; \boldsymbol{\theta}\right). \tag{3.20}$$

Having introduced Gaussian processes, their multi-task extensions, as well as GP adapters, we next focus our attention on $h_\phi$, i.e., the downstream classifier of the end-to-end learning pipeline.

## 3.3 TEMPORAL CONVOLUTIONAL NETWORKS

Temporal convolutional networks (TCNs) describe a neural network architecture that is based on convolutional neural networks (CNNs) [118], a popular class of neural networks that have contributed to significant breakthroughs in computer vision, machine translation, and audio synthesis throughout the last decade [65, 79, 151]. CNNs may be most famous for being widely adopted in various prediction tasks involving image data. However, CNNs have also been successfully applied to sequence modelling tasks, even as early as the late 80s [84]. While the term TCN was first used in Lea et al. [117], we consider a generic formulation of TCNs similar to the one presented in Bai et al. [7]. A TCN can be seen as a simple extension to a conventional 1D-CNN that fulfills three properties:

1. Causal convolutions: TCN outputs are a non-linear function of present and past sequence inputs. This imposes a notion of temporal ordering on the input data, whereas prediction outputs are not allowed to be influenced by data from the *future*.
2. Long-term memory: by employing dilated convolutions (see Definition 6), very long effective memory can be realised since the receptive field grows with $\mathcal{O}\left(2^l\right)$ for a layer at depth $l$ in the network (assuming each layer applies a dilation factor of $\delta = 2^l$).
3. Sequence to sequence: similar to an RNN, the output and each hidden layer of a TCN share the same length as the input sequence. This is achieved via the combination of causal convolutions with a padding of (kernel size $-1$) zeros before the start of the input and each hidden layer.

**Definition 6.** *For $s \in \mathbb{Z}$, let $\Omega_s = [-s, s] \cap \mathbb{Z}$. Then, for a discrete function $\chi\colon \mathbb{Z} \to \mathbb{R}$ and a discrete filter $h\colon \Omega_s \to \mathbb{R}$ of length $2s+1$, following Yu et al. [216], we define the $\delta$-dilated discrete convolution operator $*_\delta$ such that*

$$(\chi *_\delta h)(k) = \sum_{k=i+\delta\cdot j} \chi(i)\cdot h(j). \tag{3.21}$$

With $\delta = 1$, we recover the regular 1D convolution operation. In practice, we treat the values $\mathbf{x}$ of a fully-observed time series $T$ per channel as the function values $\chi(i)$ evaluated at position $i$ and refer to the length of the filter as the kernel size. A TCN is organised in residual temporal blocks, where in a given temporal block a sequence of operations (convolutions, activations, normalisations and drop out) is applied to the input and the resulting output is again added to the input (residual connection). For more details, we refer to Figure 3.2.

## 3.4 MGP-TCN: GAUSSIAN PROCESS TEMPORAL CONVOLUTIONAL NETWORKS

Having introduced the individual building blocks, we can now consider Gaussian process temporal convolutional networks (MGP-TCNs), our sepsis prediction method that combines uncertainty awareness in irregularly-spaced time series with causal dilated convolutions, a powerful inductive bias for modelling sequences and time series. In Figure 3.1, we give an overview of the entire pipeline. For a given patient encounter $i$, we observe irregularly-sampled and multivariate time series data that comprise measurements from laboratory and vital parameters. We collect these data $T_i = (\mathbf{x}_i, \mathbf{t}_i)$ in a vector $\mathbf{x}_i$ of values and a vector $\mathbf{t}_i$ of times according to Definition 4. Next, a multi-task Gaussian process (MGP) predicts a latent time series $\mathbf{z}_i$ at evenly-spaced times $\mathbf{t}_{*_i}$ while uncertainty is retained with $\mathbf{z}_i$ following a multivariate normal distribution conditioned on the observed data. Next, $\mathbf{z}_i$ is fed into a temporal convolutional network (TCN) to predict the class label $y_i$ of the patient, i.e., whether or not the patient will develop sepsis. Internally, the TCN assumes a reshaped input $\mathbf{Z}_i \in \mathbb{R}^{r \times d}$ for the imputed time series at $r$ times and of $d$ dimensions. For a given classification loss $\ell(\cdot, \cdot)$, the GP adapter framework allows us to optimise both the parameters $\boldsymbol{\theta}$ of the MGP and the parameters $\boldsymbol{\phi}$ of the TCN jointly using gradient descent (see Equation 3.18). Thus, both the imputation of the latent time series as well as the classification of the evenly-spaced, imputed time series are learned end-to-end.

Figure 3.2 provides further details about the TCN architecture. Following Section 3.3, we employ causal and dilated convolutions as organised in temporal blocks that combine two layers of convolution and normalisation with a residual connection. As for the normalisation layer for stabilising the gradients (and therefore the training), we follow Lee [119] by using a layer normalisation [6] instead of a weight normalisation [179] employed in Bai et al. [7]. In order to achieve a long effective memory size, even when using exponentially growing dilation factors, the resulting networks can become very deep, which can lead to unstable training. This is the main reason why normalisation layers are a key ingredient to TCNs.

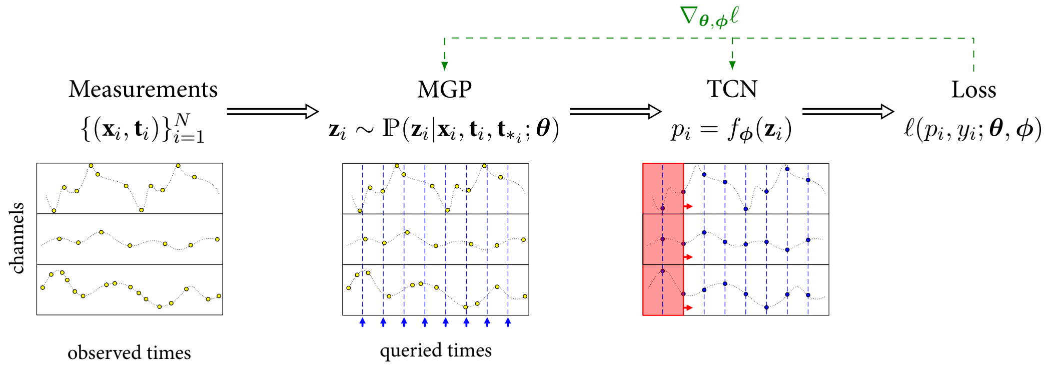

$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\ell$
Measurements
$\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$
MGP
$\mathbf{z}_i \sim \mathbb{P}(\mathbf{z}_i|\mathbf{x}_i, \mathbf{t}_i, \mathbf{t}_{*_i}; \boldsymbol{\theta})$
TCN
$p_i = f_{\boldsymbol{\phi}}(\mathbf{z}_i)$
Loss
$\ell(p_i, y_i; \boldsymbol{\theta}, \boldsymbol{\phi})$
channels
observed times
queried times

Figure 3.1: Pipeline overview of the MGP-TCN model. Irregularly-spaced time series ($\mathbf{x}_i$ refer to observed values, $\mathbf{t}_i$ to observed times) are provided to the multi-task Gaussian process (MGP). $\mathbf{z}_i$ represents the latent time series sampled at evenly-spaced query times (i.e., every hour) and follow the MGP posterior distribution (with parameters $\boldsymbol{\theta}$). Draws from $\mathbf{z}_i$ are then fed to the temporal convolutional network (TCN) (with parameters $\boldsymbol{\phi}$) which returns a prediction $p_i$. Finally, the loss function $\ell$ compares the prediction $p_i$ with the label $y_i$. The green arrow indicates that both the MGP and the TCN are trained end-to-end using the gradient of the loss. Figure recreated from [142] under retained copyrights.

We can compute the reach of the receptive field in terms of how far in the past (in terms of number of time steps) an input can affect a current output as the sum $\sum_{l=0}^{L} j2^l$ where $l$ enumerates the temporal blocks, $j$ indicates the number of stacked convolution layers within a block, and where the $l^{\text{th}}$ block employs a dilation factor $\delta = 2^l$. Even when using $j = 2$ as in our case (see Figure 3.2), that is, applying each dilation factor twice, we require a network $9$ blocks deep in order to reach inputs $1{,}000$ time steps into the past. However, due to the exponential setting, adding an additional $10$ blocks allows us to reach more than a million steps into the past, which could be interesting for learning with very long sequences as encountered, for instance, in reinforcement learning problems with video games. Now that we have introduced MGP-TCN, in the next section we outline our experiments.

## 3.5 EXPERIMENTS

We structure our experimental section in the following way: Section 3.5.1 presents the investigated dataset. Then, Section 3.5.2 specifies the sepsis prediction task, and Section 3.5.3 details the data filtering steps. In Section 3.5.4, we introduce further comparison methods. Finally, we outline our training and evaluation strategy in Section 3.5.5 and present the empirical results in Section 3.5.6.

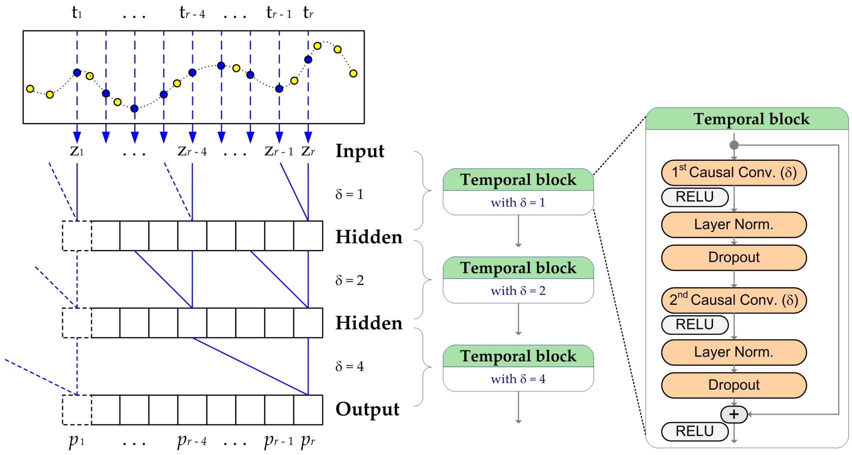


Figure 3.2: Illustration of a temporal convolutional network (TCN) processing evenly-spaced samples (values $z_k$ at evenly spaced times $t_k$ for $k = 1, \ldots, r$) of a latent time series (blue), which was predicted by the multi-task Gaussian process (MGP) as conditioned on the sparsely observed data points (yellow). The output of the TCN is denoted with $p_k$. In each temporal block, the dilation factor $\delta$ is doubled, leading to the convolution skipping an (exponentially) increasing number of outputs from the previous layer, respectively. Within a given temporal block, we apply causal $\delta$-dilated convolutions (Causal Conv. ($\delta$)), followed by a rectified linear unit (ReLU), a Layer normalisation (Layer Norm.), and a dropout layer. This sequence of operations is repeated twice, followed by a residual connection (adding the input of the temporal block), followed by a final ReLU activation. The figure was originally inspired by Bai et al. [7] and recreated from [142] under retained copyrights.

### 3.5.1 DATASET AND SEPSIS LABEL

DATASET In this chapter, we make use of the MIMIC-III dataset, version 1.4 [98]. MIMIC (Multiparameter Intelligent Monitoring in Intensive Care) is a large, freely accessible single-centre database featuring electronic health record (EHR) data from patients admitted to the critical care units of the Beth Israel Deaconess Medical Center, a large tertiary care hospital in Boston, Massachusetts. MIMIC-III includes data associated with roughly 60,000 distinct hospital admissions and approximately 45,000 unique patients (mostly adult) that was collected between the years 2001 and 2012. During the collection period, two different clinical information systems were in place: Philips CareVue Clinical Information System, which we refer to as CareVue, and iMDsoft MetaVision ICU, which we abbreviate to MetaVision [98].

SEPSIS LABEL We annotate each hour of a patient stay with a binary sepsis label following the most recent international consensus definition, Sepsis-3 [185]. As outlined in Section 2.2.1, Sepsis-3 defines sepsis as a dysregulated host response to infection. Furthermore, the authors propose a set of clinical criteria for identifying sepsis patients using the frame-

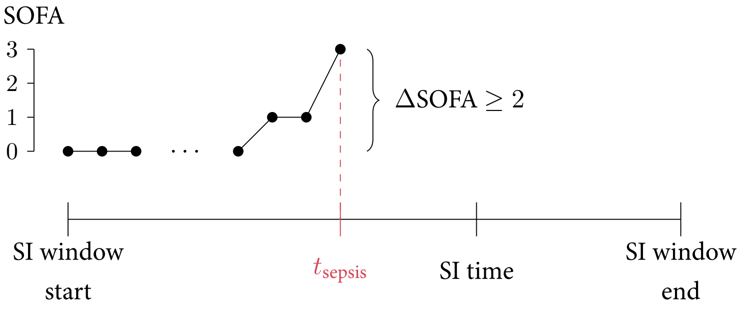


Figure 3.3: Illustration of the Sepsis-3 definition [185]. suspected infection (SI) time is determined according to Seymour et al. [181], requiring the timely co-occurrence of body fluid sampling and antibiotics administration. In a window from 48 hours before until 24 hours after SI time, the sequential organ failure assessment (SOFA) score is monitored, where an increase of at least 2 points fulfills the Sepsis-3 criteria, which was used to define the sepsis onset ($t_{\text{sepsis}}$). This figure was recreated from Moor et al. [138].

work of Sepsis-3 [185]. This involves two conditions, both of which a patient needs to fulfill: 1) a suspected (or documented) infection, and 2) signs of organ dysfunction. For the first condition, we follow the suspected infection (SI) cohort as introduced in Singer et al. [185] and further elaborated on in Seymour et al. [181]. This SI definition requires the co-occurrence of body fluid sampling and the administration of systemic antibiotics. If the culture sampling occurred first, then the drug had to be administered within 72 hours. Otherwise, if the antibiotic was given first, the body fluid sampling is required to follow within 24 hours. In this study, building on the query code of Johnson et al. [99], we use the culture sampling time to define the SI time, i.e., the onset of the suspected infection. Next, a SI window is defined as the 72 hour window surrounding the first SI time starting 48 hours before SI time and ending 24 after SI time.

To measure organ dysfunction, Singer et al. [185] propose to assess the sequential organ failure assessment (SOFA) score [204]. Specifically, an acute increase of at least 2 points in SOFA is required during the SI window, that is, between 48 hours before and 24 hours after SI time. While SOFA is computed based on the assessment of 6 vital organ systems, by design the worst values of the previous 24 hours are used, respectively [204]. In order to register an *increase* in SOFA, we repeatedly evaluate SOFA in hourly intervals. To define a sepsis onset, we then check for a 2 point increase in SOFA during the SI window and use the SOFA increase to define the onset time $t_{\text{sepsis}}$. Figure 3.3 illustrates the Sepsis-3 definition.

### 3.5.2 PREDICTION PROBLEM

Our goal is to learn a model that given a patient time series predicts its associated binary class label for sepsis. During training, we implement this as a window-based time series classification problem (see Section 2.1.3), where for a given classifier $f\colon \mathcal{T} \to \{0,1\}$, we aim to minimise the empirical risk as defined in Equation 2.3:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} \ell\Big(f\Big(T_i^{<t_{\text{sepsis}}}\Big), y_i\Big) \tag{3.22}$$

As a crucial difference to whole-series classification, here we restrict the input time series to time steps *earlier* than an instance-specific time stamp $\tau_i$ (see Equation 2.3). Since we wish to predict sepsis early, that is, based on data collected before the sepsis onset, in sepsis cases the time stamp $\tau_i$ from Equation 2.3 is set to the sepsis onset time $\tau_i = t_{\text{sepsis}}$ of patient $i$. As a default choice for setting $t_{\text{sepsis}}$ in controls without sepsis, Futoma et al. [61] used $t_{\text{sepsis}} = \infty$, i.e., no subsetting of the control time series was applied [61]. Later on in this chapter (see Section 3.5.3), we will see that $\tau_i$ needs to be chosen very carefully in time series of controls.

### 3.5.3 DATA FILTERING

COHORT SELECTION The following exclusion criteria were applied: Patients were excluded if

1. they were pediatric patients under the age of 15,
2. no chart data was available, or
3. no ICU admission or discharge time was available.

Furthermore, following the recent literature, we excluded patients logged via the CareVue system as in this logging system negative culture samplings were underreported [48]. Finally, we define an ICU encounter to be a *case* if a sepsis onset occurs during the ICU stay. Otherwise, the encounter is defined to be a *control*. To not artificially simplify the prediction task, controls were defined in an inclusive manner such that patients fulfilling only one of the two criteria (SI or SOFA) would still be deemed a control. However, to ensure that the control cohort included no patients treated for sepsis (where for instance the onset occurred before ICU admission, or where the clinical criteria of Sepsis-3 were not fulfilled), we required that control stays were not labelled with any sepsis-related ICD-9 billing codes (International Classification of Diseases, Ninth Revision).

Table 3.1: Summary statistics of the used cohort.

| **Variable** | **Sepsis Cases** | **Controls** |
|---|---|---|
| n | 570 | 5,618 |
| Female | 236 (41.4%) | 2,548 (45.4%) |
| Male | 334 (58.6%) | 3,070 (54.6%) |
| Mean time to sepsis onset in ICU (median) | 16.7 h (11.8 h) | — |
| Age ($\mu \pm \sigma$) | $67.2 \pm 15.3$ | $64.2 \pm 17.3$ |
| **Ethnicity** | | |
| White | 411 (72.1%) | 4,047 (72.0%) |
| Black or African-American | 41 (7.2%) | 551 (9.8%) |
| Hispanic or Latino | 7 (1.2%) | 147 (2.6%) |
| Other | 57 (10.0%) | 493 (8.8%) |
| Not available | 54 (9.5%) | 380 (6.8%) |
| **Admission type** | | |
| Emergency | 504 (88.4%) | 4,689 (83.5%) |
| Elective | 60 (10.5%) | 872 (15.5%) |
| Urgent | 6 (1.1%) | 57 (1.0%) |

Using all of the above criteria, we start out with 1,797 cases and 17,276 controls. As we are interested in detecting sepsis early, we exclude sepsis cases where the onset occurs during the first 7 hours into the ICU stay. We will see in Section 3.5.5 that this allows for a prediction horizon 7 hours into the future. To preserve the original class imbalance (around 10%), this final exclusion step was applied only *after* the case-control onset matching, which will be introduced in the next paragraph. Thus, our final cohort comprises 570 sepsis cases and 5,618 controls. Further summary statistics about this cohort are collected in Table 3.1. As for input variables, we included 44 vital and laboratory parameters as listed in Table 3.2 and excluded variables that were measured very rarely, i.e., fewer than 500 observations in the initial cohort (of 1,797 cases and 17,276 controls). Additionally, to address a technical limitation in one of the baselines, encounters with fewer than 10 observed measurements were excluded (for more details, please refer to Moor et al. [142, Section A.8]).

CASE-CONTROL ONSET MATCHING When framing sepsis prediction as a (window-based) time series classification task, it becomes an intriguingly impactful modelling choice to deter-

Table 3.2: Table of used input variables.

| **Vital Parameters** | |
|---|---|
| Systolic Blood Pressure | Tidal Volume Set |
| Diastolic Blood Pressure | Tidal Volume Observed |
| Mean Blood Pressure | Tidal Volume Spontaneous |
| Respiratory Rate | Peak Inspiratory Pressure |
| Heart Rate | Total Peep Level |
| SpO2 (Pulsoxymetry) | O2 flow |
| Temperature Celsius | FiO2 (Fraction of Inspired Oxygen) |
| Cardiac Output | |
| **Laboratory Parameters** | |
| Albumin | Blood Urea Nitrogen |
| Bands (Immature Neutrophils) | White Blood Cells |
| Bicarbonate | Creatine Kinase |
| Bilirubin | Creatine Kinase MB |
| Creatinine | Fibrinogen |
| Chloride | Lactate Dehydrogenase |
| Sodium | Magnesium |
| Potassium | Calcium (free) |
| Lactate | pO2 Bloodgas |
| Hematocrit | pH Bloodgas |
| Hemoglobin | pCO2 Bloodgas |
| Platelet Count | SO2 Bloodgas |
| Partial Thromboplastin Time | Glucose |
| Prothrombin Time (Quick) | Troponin T |
| INR (Standardized Quick) | |

mine exactly which time windows of a patient are used. This problem has been characterised before with the prediction of in-hospital mortality and hypokalemia [182]. For the prediction of sepsis, cases have been aligned to control patients by assigning a matched pseudo-onset in control patients, i.e. patients without sepsis. For instance, Futoma et al. [62] demonstrated a drastic drop in predictive performance (almost halving the area under the precision-recall curve (AUPRC)) of the same method on the same dataset by merely changing the control onset from discharge time [61] to a matched onset time [62]. In terms of our formalisation of the prediction problem, the first choice of matching, that is, setting the pseudo-onset to the discharge time refers to Equation 3.22 with setting $t_{\text{sepsis}} = \infty$ for controls [61]. The authors noted in the follow-up paper, that this choice may have rendered the problem too easy, as controls shortly before discharge would be expected to be in an overall better health state than a critically ill patient developing signs of sepsis [62]. Thus, to address this, they matched a given case to 4 controls (roughly preserving their prevalence of around 20%) by assigning a pseudo-onset (they refer to it as "prediction time") at the same fraction of the stay duration as the real sepsis onset occurred in the matching case [62].

Motivated by this, and to avoid an overly simplified classification task, here we also employ case-control onset matching. However, given our class imbalance, we match each case to 10 controls and assign the *absolute* time of sepsis onset since admission (as opposed to a relative onset time calculated via the fraction of the stay at which sepsis onset occurred) to derive matched control onsets at the same number of hours after admission. We changed this relative matching to an absolute matching since we observed that cases and controls did not necessarily show a similar length of stay, which could have led to biases in the alignment that a sufficiently powerful classifier could have plausibly exploited. Finally, for each case and matched control, we extracted up to 48 hours of in-ICU input data preceding the (matched) sepsis onset and succeeding the time of ICU admission.

### 3.5.4 COMPARISON METHODS

We compare MGP-TCN to a set of comparison methods. This includes 1) Gaussian process recurrent neural networks (MGP-RNNs), 2) temporal convolutional networks (TCNs), and 3) dynamic time warping $k$-nearest neighbors (DTW-$k$NNs). MGP-RNN represents the first multi-task Gaussian process adapter model for predicting sepsis [61, 62]. As MGP-TCN was motivated by MGP-RNN (we empowered the uncertainty aware framework with the inductive bias of TCNs), we deem the comparison to this baseline most interesting. Next, as an ablation of the MGP component of MGP-TCN, we compare to a TCN that works with an evenly-spaced time series that was manually preprocessed using carry-forward imputation (for more details, please refer to next paragraph). Finally, we also employ a

DTW-$k$NN classifier, a powerful data mining method based on dynamic programming that has shown competitive time series classification performance [45]. Also, in contrast to many other (non-deep learning) classifiers, DTW can be directly applied to time series of varying lengths. To our knowledge, DTW-$k$NN has not been applied to sepsis prediction before. For DTW-$k$NN, we reuse the same carry-forward imputation that was created for the TCN model. DTW distance matrices were computed for each time series channel such that channel-wise $k$NN classifiers were ensembled via soft votes, i.e., by averaging the prediction score over all channels (as opposed to a hard vote, where the most frequently predicted class would be chosen).

IMPUTATION SCHEMES Here, we give more details about the imputation scheme that was used in the comparison methods that did *not* employ an MGP that could directly process the irregular time series. As the MGP was queried every hour of the patient stay, for maximal comparability we collected the raw time series in hourly bins, where each bin was assigned the mean as computed from all available observations inside the bin. Then, we filled empty bins using the value of the last non-empty one (carry-forward imputation). Any remaining missing entries at the start of the time series were imputed using the mean of the respective channel as computed on the entire training dataset.

### 3.5.5 EXPERIMENTAL SETUP

TRAINING We randomly divide the data into 80% data for training, and each 10% for validation (to tune the hyperparameters) and for testing (to report the final performance evaluation). This splitting strategy was applied in three independent iterations to enable an assessment of performance variability. In each iteration of data splitting, the respective training set was used to estimate sample mean and standard deviation of each channel in order to then apply channel-wise $z$-scoring to the entire dataset. This means that a time series value $x_{ij}$ observed at time $t_i$ for the channel (or dimension) $j$ is mapped to $z_{ij} = \frac{x_{ij}-\mu_j}{\sigma_j}$ where $\mu_j, \sigma_j$ indicate the training set-wide mean and standard deviation of channel $j$.

For hyperparameter tuning, to avoid expensive evaluations over a large grid of hyperparameters, we instead used a Bayesian optimisation framework [80] and ran 20 calls (each corresponding to one hyperparameter configuration) for each method and split iteration. Hyperparameters and model checkpoints were selected to maximise area under the precision-recall curve (AUPRC) as computed on the validation split. For each method and split iteration, the best model checkpoint was then applied to the respective test set for evaluation. To keep the parametrisation of the deep models comparable, we constrained the hyperparameter configurations of the deep models such that the resulting number of trainable parameters ranged between 20,000 and 500,000 parameters. To prevent overfitting, we stopped training when the validation AUPRC did not improve over 5 epochs. In order to make it feasible to tune deep neural networks over multiple iterations of random splitting (also referred to as Monte Carlo cross validation), we restrict each run to train for at most 2 hours. Furthermore, to counteract underfitting due to this temporal constraint, each best parameter configuration was retrained for a longer period (50 epochs for the MGP-based methods that would only train for 5 to 15 epochs, otherwise). For DTW-$k$NN, we precomputed the distance matrices on the entire dataset to evaluate different values of $k \in \{1, 3, \ldots, 13, 15\}$ on the same validation set as used in the other methods. For further details regarding the hyperparameter search, please refer to Moor et al. [142, Section A.5 and Table A.2].

Evaluation To evaluate classification performance, due to a significant class imbalance (only 9.2% sepsis cases) we focus on area under the precision-recall curve (AUPRC). In order to conform with the recent sepsis prediction literature, we also report area under the receiver-operating-characteristic curve (AUROC), albeit being aware of the shortcomings of this measure when dealing with imbalanced datasets, i.e., that AUROC becomes more meaningless with increasing class imbalance [178]. As we ultimately care about the *early* identification of sepsis, we perform an horizon analysis where we assess classification performance going back in time up to 7 hours before onset and matched onset, respectively. Specifically, for each hour up to 7 hours before onset we restrict the model input data to only contain data up until the current hour to arrive at a measure of how good would the model have predicted sepsis $n$ hours before the onset. To give an example, to evaluate the prediction horizon 4 hours in advance, all fitted models (including the MGP imputation) were only provided patient data up until 4 hours before sepsis onset and matched onset in order to make a prediction. It is worth emphasizing that the models were not retrained for specific time horizons in order to squeeze out even better numerical results, since the goal of this study was to learn a *single*, potentially deployable model, and to assess how well it could detect sepsis when querying predictions during early windows before sepsis by gradually removing information before sepsis onset.

### 3.5.6 Results

Figure 3.4 displays the classification performance over the different prediction horizons. On the $x$-axis, the prediction horizon, i.e., the number of hours the prediction step precedes sepsis onset, is shown. This is plotted against two measures of classification performance on the $y$-axis: 1) area under the precision-recall curve (AUPRC), an evaluation metric that is suitable for classification under class imbalance (left panel), and 2) area under the receiver-

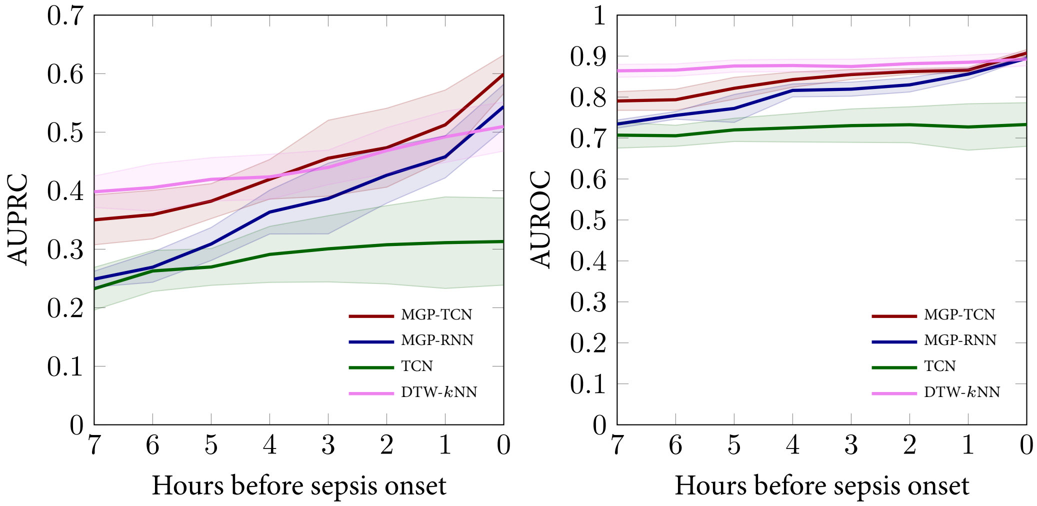


Figure 3.4: All methods were tuned and evaluated in terms of area under the precision-recall curve (AUPRC). Additionally, we display area under the receiver-operating-characteristic curve (AUROC).

operating-characteristic curve (AUROC), an evaluation metric that is less informative under class imbalance, however widely adopted in the literature of clinical prediction models (right panel). The error bands indicate the observed standard deviation when computing the metrics for each iteration of splitting. When considering our main evaluation measure, AUPRC, we observe that MGP-TCN as well as the DTW-$k$NN ensemble, both novel methods for sepsis prediction, perform favourably compared to the previous state-of-the-art sepsis detection method MGP-RNN. Considering prediction horizons earlier than $4$ hours ahead of sepsis, both MGP-TCN and DTW-$k$NN outperform MGP-RNN by a substantial margin. Approaching sepsis onset, the performance curves of these three methods intertwine with overlapping variability estimates. At $4$ hours before onset, the AUPRC mean curve of the two best performing methods, MGP-TCN and DTW-$k$NN, cross such that earlier than $4$ hours in advance, DTW-$k$NN shows a higher mean AUPRC, whereas later than $4$ hours in advance MGP-TCN displays the higher mean AUPRC. Nevertheless, the error bands of these two methods are overlapping in all horizons except for hour $0$. Furthermore, we observe that the TCN does not show a competitive performance at any prediction horizon. Finally, we find that methods that work with evenly-spaced time series (here after preprocessing with carry-forward imputation), i.e., DTW-$k$NN and TCN, compared to the uncertainty-aware methods upon approaching sepsis onset show a flatter increase in the performance curves.

## 3.6 DISCUSSION

In this chapter, we presented MGP-TCN, a novel model for predicting sepsis that combines uncertainty-awareness of MGP adapters with causal and dilated convolutions of TCNs. To develop and empirically test this method, we have built the first publicly available sepsis early prediction pipeline that includes hourly sepsis annotations as derived from the Sepsis-3 definition. In our experiments, we compare MGP-TCN against its predecessor MGP-RNN, which previously achieved state-of-the-art performance in sepsis prediction [61]. Furthermore, we compared against a classic TCN as an ablation to evaluate if MGP-TCN actually benefits from leveraging the raw, irregularly-sampled time series with an MGP adapter. Finally, to juxtapose these three deep learning approaches with a non-deep learning method, we also investigate a channel-wise ensemble of DTW-$k$NN classifiers. We found that both MGP-TCN and DTW-$k$NN exhibit competitive performance and outperform MGP-RNN. Our findings in Figure 3.4 entail that MGP-TCN improves performance over both the MGP-free classic TCN as well as the TCN-free MGP-RNN baseline, thereby showcasing that recent progress in sequence models (using causal dilated convolutions) can be successfully transferred to medical time series datasets with incomplete and irregularly spaced observations. Interestingly, we observe that DTW-$k$NN performs surprisingly well, outperforming several of the deep learning methods in terms of AUPRC. However, we caution against overinterpreting this specific finding, as it may partially be an artefact of the limited sample size which could restrict the performance of the deep learning approaches that typically excel in the very large sample size regime.

SCALING BEHAVIOUR While for the investigated deep learning methods the runtime cost of predicting a single instance is constant with increasing sample size (i.e., number of patients), for DTW-$k$NN this is generally not the case making it much harder to scale this method to large cohort sizes. When using exact inference, drawing samples from the MGP posterior involves a Cholesky decomposition of the $(d \cdot t_i) \times (d \cdot t_i)$ covariance matrix $\mathbf{\Sigma}_i$ of instance $i$ (as computed in Equation 3.13) which for $t_i = \text{len}(T_i)$ observed time steps and $d$ channels (or dimensions) leads to a runtime of $\mathcal{O}\Big((d \cdot t_i)^3\Big)$. However, since the MGP imputation is computed per instance individually (under shared hyperparameters), the cost is cubic only in the dimensionality and in the number of time steps of the current time series $T_i$, making it possible to scale this method to larger sample sizes. If necessary, this cost can be reduced by approximating the posterior, for instance with the Lanczos method where the runtime is cubic only in a parameter that is chosen to be a small constant [125]. By contrast, the DTW-$k$NN ensemble is much harder to scale. Even though as a lazy learner it has a negligible runtime during training ($\mathcal{O}(1)$), the runtime complexity during prediction is substantial. For an unseen instance, the DTW distance to all $n$ training instances needs to be computed where one pair-wise distance computation already costs $\mathcal{O}\left(d(t_i)^2\right)$ with $t_i$ again referring to the number of time steps (although in our case, DTW-$k$NN processed hourly sampled, evenly-spaced time series). In total, already for computing the distances for a single new instance upon prediction time this involves a cost of $\mathcal{O}\left(nd(t_i)^2\right)$. For our dataset of mid-range size, predicting at a single horizon already requires the alignment of hundreds of millions of pairs of time series (univariate), followed by the storing of the distances, which can lead to significant runtime and memory overheads.

Limitations and future work In the following, we discuss limitations to this study as well as opportunities for future work. First, the empirical performance assessment was restricted to a single dataset, MIMIC-III, whereas an external validation of the models could not be conducted due to the lack of an accessible and annotated validation dataset. We contacted the authors of several related works: Futoma et al. [61] could not make the patient data used in their paper available. Furthermore, the authors of the papers introducing and validating the "InSight" method for sepsis prediction would not share their code nor the queried data due to proprietary interests, even though in their papers they used MIMIC-III, a publicly available dataset distributed under an Open Database License that requires derived data that was publicly used to be made reproducible [24, 48]. Already the labelling of a single dataset was very labour-intense due to the complexity of interpreting and implementing the Sepsis-3 definition. Therefore, for as long as labelling code and data keeps being unavailable, the sepsis prediction literature will continue being in dire need of accessible and annotated datasets for validation. In this study, we retrospectively investigated whether the hours preceding sepsis onset carry signals predictive of the inbound sepsis onset by comparing said time windows with time windows in control patients. While such a binary classification setting is interesting in that it reveals whether there are actually signals predictive of sepsis, this approach is retrospective by design, i.e., we start out by knowing when sepsis starts and investigate how early in advance the model could have predicted it. However, when deploying a sepsis early warning system, it would prospectively monitor patients and repeatedly output predictions. This limitation is in line with previous work on clinical prediction models that found that temporally aligning the prediction task with the clinical event of interest may be insufficient for evaluating a model with regard to clinical usability [182].

To more closely reflect a deployment scenario, in Chapter 4, we will encounter an online monitoring setting already during the (retrospective) training and evaluation of the models. However, the best performing methods in this study will require some further considerations in order to train them for and apply them in an online monitoring setting. For instance, a direct application of our used DTW-$k$NN approach would imply that for each new observed measurement the distance of a patients time series to all training time series needs to be updated or at least partially recomputed, which would be very costly either in terms of runtime or memory. As extensions of DTW to an online setting are being investigated [152], online extensions of DTW-$k$NN could become an exciting topic for future work. Moreover, it will be interesting to reformulate MGP-TCN for an online scenario from the viewpoint of local GPs [147, 218]. In the subsequent chapter, several open problems remaining after this first study will be investigated, including external validations, an online prediction scenario, as well as model explanations.

# 4 Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning

In this chapter, we present a multi-centre study for the development and validation of sepsis early warning systems using machine learning. The content is based on the following preprint which is currently under review:

M. Moor[†], N. Bennett[†], D. Plečko[†], M. Horn[†], B. Rieck, N. Meinshausen, P. Bühlmann, and K. Borgwardt. "Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning". *arXiv preprint arXiv:2107.05230*, 2021

We start by putting this chapter into context with the previous sections of this thesis. First, in Section 2.2, we introduced sepsis and sketched how the early recognition of this syndrome is both clinically relevant as well as challenging, which given an abundance of monitoring data leads to an interesting machine learning prediction problem. Then, in Chapter 3, we elucidated the related literature and developed MGP-TCN a novel method for sepsis prediction, and empirically investigated it together with several deep learning and data mining methods in a retrospective time series classification task on the MIMIC-III dataset. While we observed convincing results in terms of predictive performance, the study outlined in the previous chapter faced several limitations that the current chapter is going to address. This chapter is organised as follows: In Section 4.1, we elucidate the current gap in the literature and summarise the contributions of this chapter. Next, in Section 4.2 we detail the study design, the prediction problem, the employed prediction methods, the experimental setup, as well as our evaluation strategy. Finally, in Sections 4.3 and 4.4, we present the empirical results and discuss the impact, scope, and limitations of our findings.

## 4.1 Introduction

For decades, sepsis has persisted to be a dominant cause of mortality and morbidity [47, 94, 103]. Even though an early identification would improve prognosis by enabling timely disease management [54], a long sought-after gold standard for the early diagnosis of sepsis is still missing.

Given a wealth of routinely collected laboratory and monitoring patient data, the prediction of sepsis from this data has become an attractive machine learning problem. However, compared to more conventional clinical prediction targets, such as mortality, length of stay, or time to hospital readmission, sepsis describes a complex and heterogeneous clinical entity that is challenging to consistently define. Even though the last few decades have witnessed several refinements and reevaluations of the international consensus definitions for sepsis [122, 185], the problem of how to best define sepsis and sepsis-related outcomes is far from being solved. This is also reflected in the current literature where a bewildering number of definitions and ad-hoc approaches are used to determine the time of sepsis onset [143].

Even though recent years brought forth a considerable body of literature investigating the early predictability of sepsis using machine learning (refer to Fleuren et al. [56] and Moor et al. [143] for a systematic overview), there are several factors that hinder a straight-forward comparison of these approaches, rendering a direct juxtaposition of numerical results largely futile. Besides heterogeneous sepsis definitions, previous studies have also framed their prediction problem in heterogeneous ways. For instance, pseudo-onsets in matched controls were either assigned to the discharge time [61], to a time after a relative proportion of the hospital stay [62], to a time after an absolute number of hours into the stay [142], or as in the most cases, not explicitly reported at all [143].

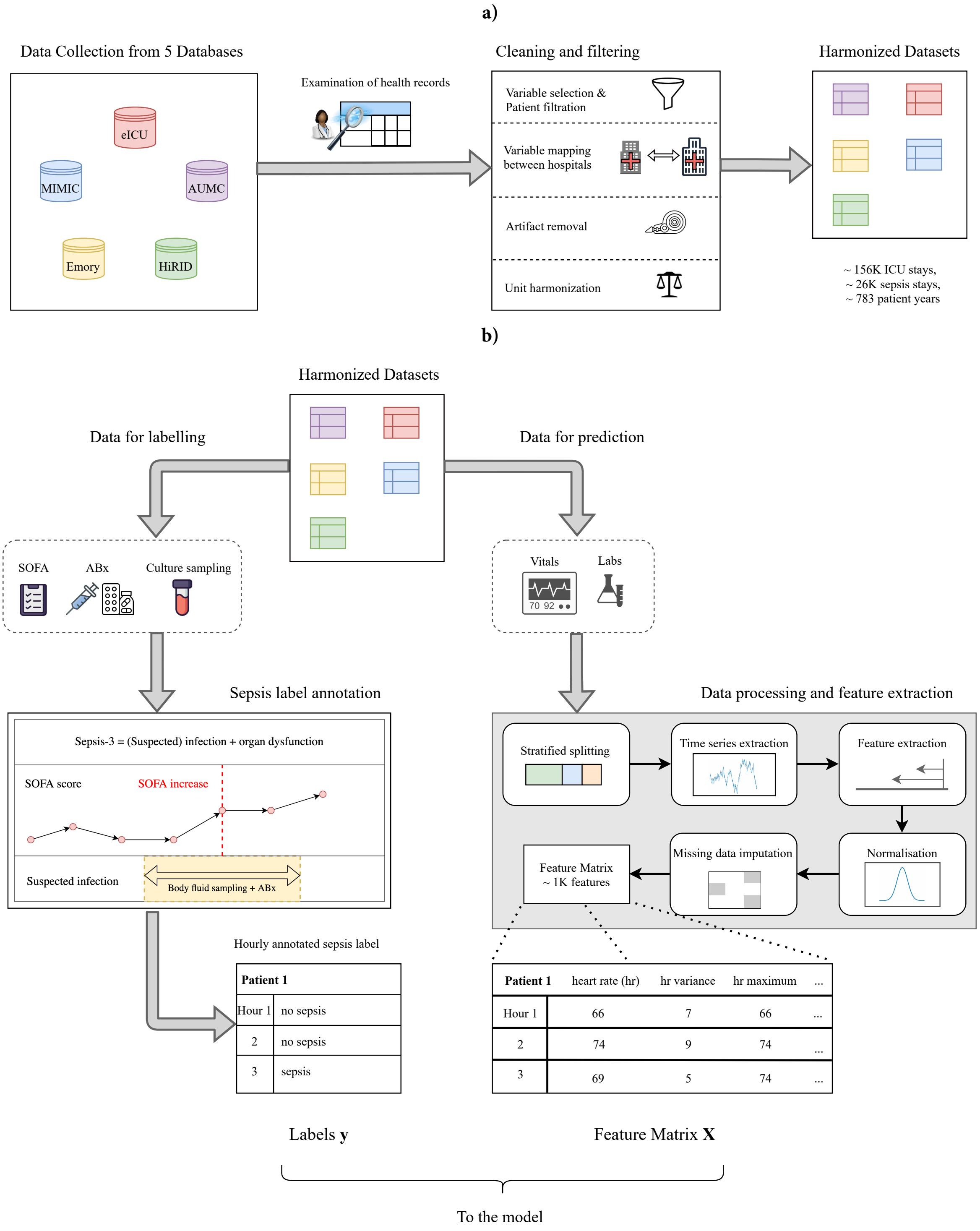
The limited comparability between existing studies is further exacerbated by only a small fraction of studies having employed an external validation [143]. As a foundational issue underlying these phenomena, there is a lack of consistently annotated data originating from different centres. In fact, currently the majority of publications predicting sepsis in the ICU were developed on the MIMIC-III dataset, plausibly due to ease of access and the high quality of the data [143]. Lacking access to open-access, annotated data for validation is a core driver of the problem that most sepsis prediction studies lack external validations. On top of that, a recent study found a widely deployed proprietary sepsis prediction model to perform surprisingly poorly when validated externally [211], which begs the question whether proprietary prediction models ought to be better validated as opposed to being rushed into deployment. Currently, however, this can not be easily implemented, since multi-centre, annotated validation data is lacking. Motivated by these circumstances, in this chapter we present a multi-centre study unifying data from five EHR databases to conduct the first international external validation of sepsis prediction using machine learning. Specifically, the contributions of this study are as follows:

- We harmonise, clean, and filter data from five databases to create the first international benchmark comprising over $150,000$ ICU stays.
- We derive hourly sepsis labels using the Sepsis-3 definition [185].
- We develop sepsis prediction models using state-of-the-art machine learning methods and compare them against several clinical baselines.
- We devise an evaluation strategy that captures both alarm accuracy and earliness.
- Using this unique benchmark, we perform an extensive external validation across centres, and for the first time across nations and continents.

## 4.2 METHODS

### 4.2.1 STUDY DESIGN AND DATA SOURCES

We conducted an observational and retrospective study employing multiple centres. First, this involved the creation of a multi-centre ICU cohort comprising sepsis patients and controls. For this, Figure 4.1 gives an initial overview of the data flow and the preprocessing steps that we further detail in the subsequent sections. Next, we developed, internally validated, and externally tested sepsis warning systems that aim to detect sepsis during the acute first week of an ICU stay. The investigated study cohort comprises ICU patients across three countries and two continents as collected in the following databases (versions provided in parentheses whenever available): i) HiRID [95] (1.1.1), ii) AUMC [196] (1.0.2), iii) MIMIC-III [98] (1.4), iv) eICU [158] (2.0), v) and Emory [166]. In all datasets, the Sepsis-3 definition was used. The Emory dataset was made available in a preprocessed stage as part of the 2019 PhysioNet Computing in Cardiology Challenge [166]. Since the public dataset was not accompanied with the necessary information to derive the label, and since this challenge data was already equipped with a sepsis label based on Sepsis-3, for this dataset we used the existing Sepsis-3 annotations as provided in the challenge data. Furthermore, since the published PhysioNet challenge data is composed of two sets corresponding to data from Emory and MIMIC-III, to prevent redundancy in our cohort, here we only use the Emory set. The data was for the most part collected during the last decade: AUMC from 2003 to 2016, eICU between 2014 and 2015, MIMIC-III from 2001 to 2012, HiRID from 2008 to 2016, and Emory “during the last decade” (while the Emory data was made available in early 2019) [166]. Notably, the data underlying this multi-centre cohort was collected *before* the COVID-19 pandemic started.



| Patient 1 | |
|---|---|
| Hour 1 | no sepsis |
| 2 | no sepsis |
| 3 | sepsis |

| Patient 1 | heart rate (hr) | hr variance | hr maximum | ... |
|---|---|---|---|---|
| Hour 1 | 66 | 7 | 66 | ... |
| 2 | 74 | 9 | 74 | ... |
| 3 | 69 | 5 | 74 | ... |

Figure 4.1: An illustration of the preprocessing pipeline. Panel a): We collected, cleaned, and harmonised ICU data from five EHR databases. Panel b): Next, on the left we illustrate how sepsis labels were derived following Sepsis-3. On the right side, we visualise the preprocessing steps that were applied to extract features that are used for prediction. Figure recreated from [138].

In this study, we continuously monitored laboratory measurements and vital parameters while considering demographic patient information. As for input variables, by design we restricted ourselves to variables that were i) plausibly related to sepsis[1] ii) consistently observed, and iii) not a direct indicator of sepsis treatment, in order to prevent spurious situations where a model would merely wait for the attending clinician to detect and treat sepsis. Such a dependence could become problematic upon deployment, as in the worst case both the clinician and the early warning system would await each other's actions. To account for this last criterion, we excluded therapeutic variables such as administrations of intravenous fluids, antibiotics, and vasopressors from the collection of input variables used for predicting sepsis. To facilitate interoperability between the available datasets, we undertook harmonisation steps along two axes: First, we harmonised the temporal resolution by resampling all datasets to hourly bins by reporting the median of each bin. Second, based on the above mentioned criteria, and by trading off variable overlap between the datasets with still having a sizeable number of variables, we devised a consensus set of 59 time series variables and 4 static covariates (see Table 4.1).

Exclusion criteria We applied to following filtering steps to create our study cohort:

i) pediatric patients under the age of 14 were excluded, and

ii) hospitals that showed a very low prevalence in patients fulfilling Sepsis-3 ($< 15\%$) were removed as they would have lead to false-negative control patients due to poor data availability.

Furthermore, fulfilling at least one of the following criteria led to exclusion. Specifically, we excluded an ICU stay if

iii) it showed a length of stay of less than 6 hours,

iv) it contained fewer than 4 distinct hourly bins with measured observations,

v) there was a missing data window longer than 12 hours, and

vi) sepsis onset occurred before 4 hours into ICU stay, or later than 168 hours after ICU admission.

We further illustrate the individual steps and the corresponding number of excluded patient stays in Figure 4.2.

---

[1] Even indirect relations could be helpful. For instance, certain demographics such as weight or height may not be directly related to sepsis but could still potentially carry information about the constitution of the patient that could affect how vital signs are to be interpreted and could have implications about the intubatability of the patient.

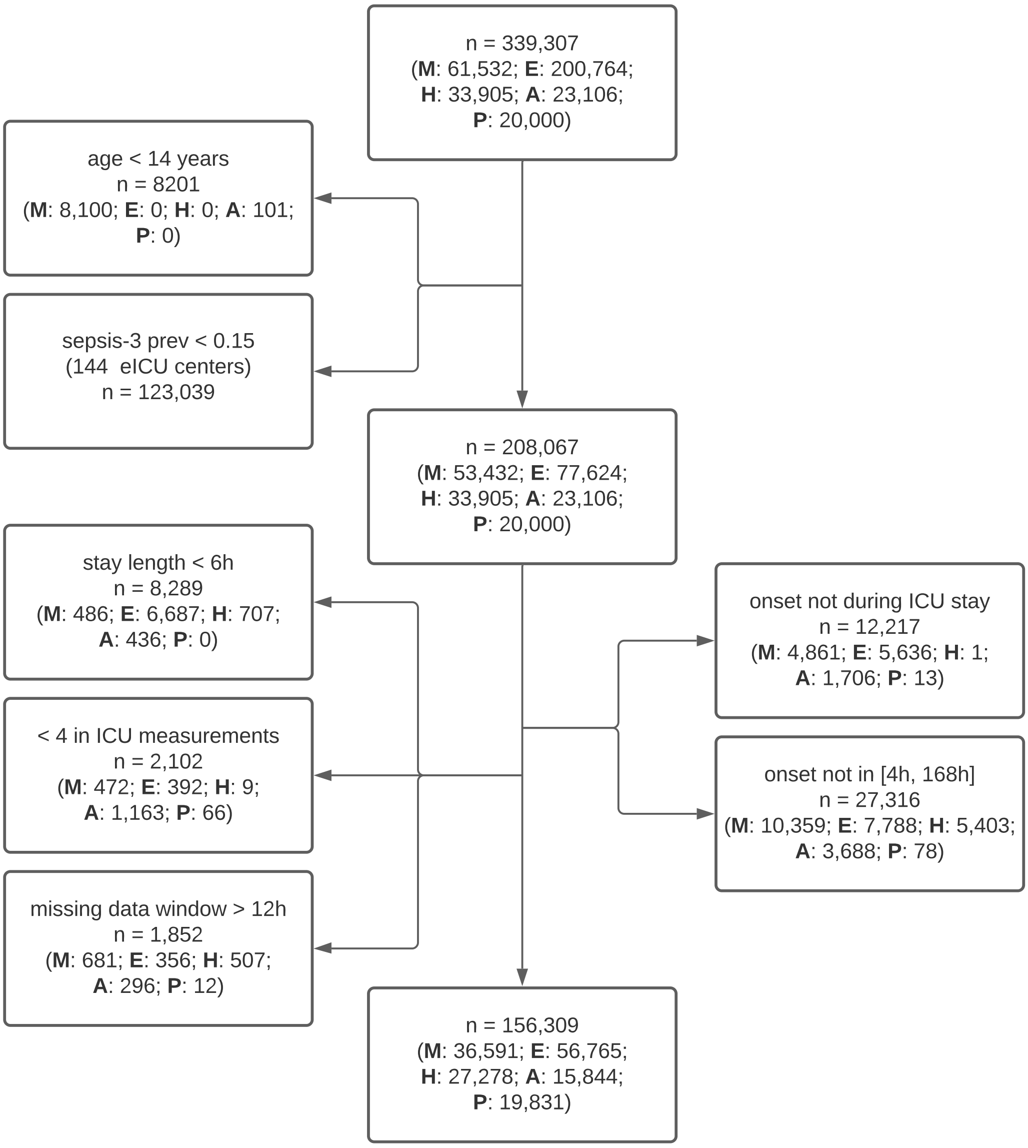


Figure 4.2: Study flowchart to illustrate the number of excluded patient stays due to the applied filtering steps. Figure recreated and adapted from [138].

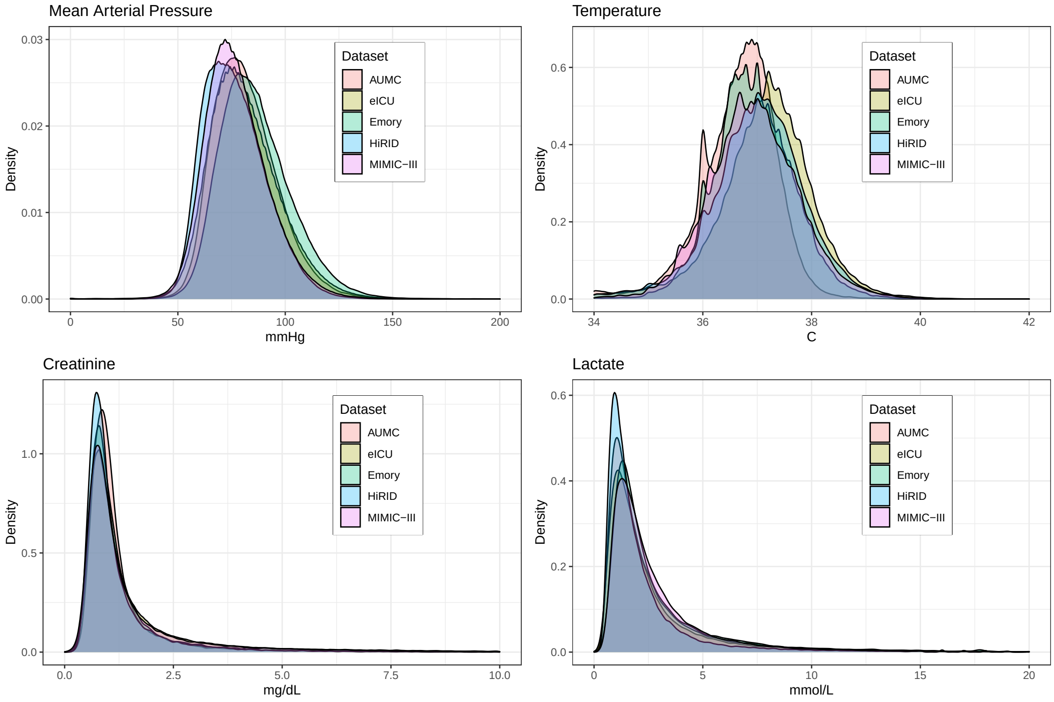


Figure 4.3: Illustration of harmonised data distributions exemplified for four variables.

Data harmonisation We manually inspected the included variables and plotted distributions of observed values for all datasets. This was particularly necessary in order to harmonise units across the different data sources. In Figure 4.3, we display the distribution of observed values for four different variables, and find that they neatly align across the datasets after unit harmonisation.

In summary, the conducted preprocessing steps were designed to maximise interoperability and harmonisation across the datasets. For instance, extracting hourly bins of measurements effectively leads to a lower data resolution in the HiRID dataset, where many variables are recorded every 2 minutes. However, in the original paper of this dataset, we found that the transfer of models to datasets with hourly resolution is facilitated when adopting the hourly frame already when training on HiRID [95]. Thus, there is an inherent trade-off between leveraging the wealth of a dataset versus preparing it in an interoperable and generalisable manner. Furthermore, all preprocessing steps of the input data may be applied as real-time transformations, which is relevant when considering deployment scenarios.

### 4.2.2 Outcome and prediction problem

Outcome definition In this study, the onset of sepsis as defined by the Sepsis-3 definition was considered the primary outcome of interest [185]. Sepsis-3 requires the fulfillment of two clinical criteria: i) a suspected infection (SI), defined as the co-occurrence of systemically administered antibiotics[2] and sampling of body fluid cultures, and ii) signs of organ dysfunction as determined by an increase of at least two points of the sequential organ failure assessment (SOFA) score.

We followed Seymour et al. [181] to implement the SI definition: if antibiotics were administered first, the body fluid sample needed to be obtained within the next 24 hours. In the other case, if culture sampling preceded the antibiotics, the antibiotics were required to be ordered within the following 72 hours. The earlier of the two events was used to define the SI time. Next, we defined the SI window from 48 hours before until 24 hours after SI time [181]. During this window, an increase of SOFA by at least two points defined the time of sepsis onset. The SOFA score was extracted as originally proposed by Vincent et al. [204]. To prevent false-low values of the Glasgow Coma Scale (GCS), we set the GCS score in sedated patients to 15, i.e., its maximal value. Additionally, the 24-hour urine output variable (part of the kidney function component of SOFA) was only evaluated starting at 12 hours after ICU admission, where up until hour 24 the values were properly scaled to approximate a 24-hour estimate.

In two datasets, the original SI definition was hard to implement: for eICU, only a small number of body fluid records were reported; for HiRID, no body fluid samples were reported at all. Therefore, on those two datasets, we employed an alternative definition to identify suspected infections. Specifically, for the alternative SI definition we required multiple antibiotics to be administered simultaneously. To test the validity of the alternative SI definition, we compared it against the original SI definition on the AUMC and MIMIC-III datasets, where both definitions can be derived. In Figure 4.4, we show Venn diagrams illustrating the overlap between the two definitions in terms of number of ICU stays fulfilling each definition, where the original SI definition is indicated as the combination of fluid sampling with antibiotics (ABX) and the definition based on multiple antibiotics is indicated as multi-ABX.

In Panel 4.4a, when comparing the two variations of SI on MIMIC-III, we observed a Jaccard similarity[3] of 0.69. Panel 4.4b shows a smaller overlap for AUMC with a Jaccard similarity of 0.42, however most patients of the original SI definition are included in the alternative definition. Due to a frequent use of prophylactic antibiotics in surgical patients,

---

[2] Here, "systemic" refers to oral or parenteral applications, as opposed to topical applications.

[3] The Jaccard similarity of two finite sample sets $A$, $B$ is defined as the ratio of the cardinalities of the intersection and the union between $A$ and $B$, i.e., $J(A, B) = \frac{A \cap B}{A \cup B}$.

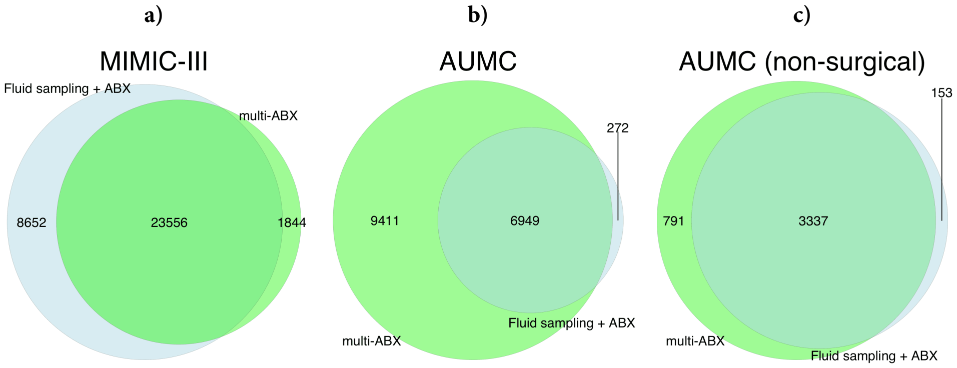


Figure 4.4: Venn diagrams to compare the two alternative definitions for a suspected infection. Panel **a)** shows a large overlap of the two definitions on the MIMIC-III dataset. In Panel **b)**, we found a considerably smaller overlap on the AUMC dataset. However, the AUMC dataset was the only used dataset which predominantly consists of surgical patients, in which a prevalent use of prophylactic antibiotics could explain the small overlap with the original SI cohort. In Panel **c)**, confirming this hypothesis, we find a large overlap of the two definitions for the non-surgical cohort in AUMC. Figure recreated from [138].

and since AUMC, among the used datasets, is the only one representing a foremost surgical cohort, we hypothesised that this could explain the lower overlap in AUMC. Indeed, when performing the comparison only on the non-surgical cohort of AUMC (see Panel 4.4c), we again observe a strong overlap with a Jaccard similarity of 0.78.

PREDICTION PROBLEM Given 59 sequentially observed laboratory and vital parameters together with 4 demographic covariates, at each hour of an ICU stay we aim to predict, whether a sepsis onset occurs during the next 6 hours. In Figure 4.5, we illustrate how we implemented this online prediction scenario during training. For controls, each hour is assigned the label 0. In sepsis cases, we start with the hourly label 0 and six hours before sepsis onset it switches to 1. After onset, we allow for 24 hours of further data to maximise the chance of encountering signals indicative of sepsis during training. As outlined in Section 4.2.1, we consider sepsis onsets that occur at most one week, or 168 hours, into the ICU stay. Accounting for the 24-hour window included after sepsis onset, to achieve comparable time series lengths, we only consider the first $168 + 24 = 192$ hours of ICU stay for control patients. Our sepsis label, which is based on Sepsis-3, partially relies on the SOFA score which incorporates treatment information (such as vasopressors), which is relevant to assess organ dysfunction, and therefore sepsis. However, as outlined in Section 4.2.1, by design we are interested in models which do not rely on therapeutic variables. Therefore, even though SOFA would be predictive of a future Sepsis-3 event, we did not directly include SOFA as an input feature that can be used for prediction. We give more details about all the variables used for

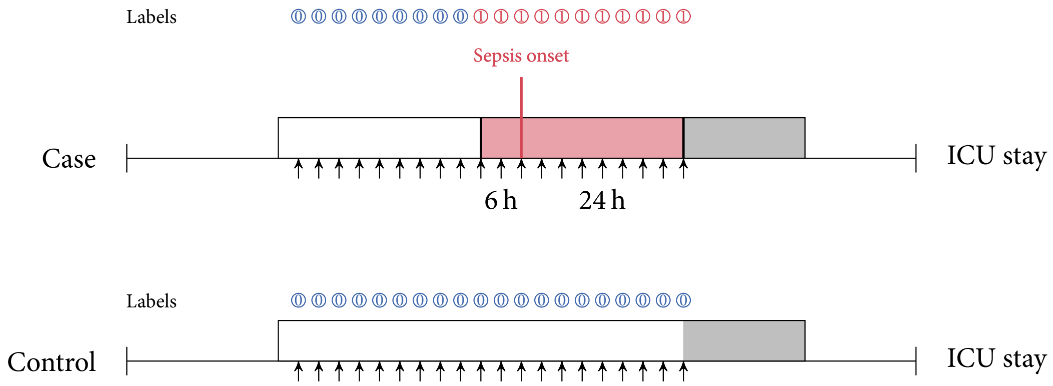


Figure 4.5: Overview of the prediction problem illustrated for a single sepsis case and control. The vertical red line indicates the sepsis onset. We assign time steps 6 hours before up until 24 hours after onset the label 1, incentivising a positive prediction shortly before sepsis onset. Positively labeled time steps after onset are included during training as they could potentially contain signals indicative of sepsis. As we focused on sepsis onsets during the first week in ICU and due to the 24-hour follow up after onset in cases, to achieve a comparable maximal length of ICU stay, data after $168 + 24$ hours, i.e. eight days into ICU stay is discarded (grey bar). Figure recreated and adapted from [138].

prediction in Table 4.1 and further detail features that were extracted from those variables in Section 4.2.4.

Table 4.1: Variables provided to the models for predicting sepsis. For each dataset, we indicate the available variables. MIMIC stands for MIMIC-III.

| Name | Description | MIMIC | eICU | HiRID | AUMC | Emory |
|---|---|---|---|---|---|---|
| age | patient age | ✓ | ✓ | ✓ | ✓ | ✓ |
| alb | albumin | ✓ | ✓ | ✓ | ✓ | ✗ |
| alp | alkaline phosphatase | ✓ | ✓ | ✓ | ✓ | ✓ |
| alt | alanine aminotransferase | ✓ | ✓ | ✓ | ✓ | ✗ |
| ast | aspartate aminotransferase | ✓ | ✓ | ✓ | ✓ | ✓ |
| basos | basophils | ✓ | ✓ | ✗ | ✓ | ✗ |
| be | base excess | ✓ | ✓ | ✓ | ✓ | ✓ |
| bicar | bicarbonate | ✓ | ✓ | ✓ | ✓ | ✓ |
| bili | total bilirubin | ✓ | ✓ | ✓ | ✓ | ✓ |
| bili_dir | bilirubin direct | ✓ | ✓ | ✓ | ✓ | ✓ |
| bnd | band form neutrophils | ✓ | ✓ | ✓ | ✓ | ✗ |
| bun | blood urea nitrogen | ✓ | ✓ | ✓ | ✓ | ✓ |
| ca | calcium | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 4.1: Variables provided to the models for predicting sepsis. For each dataset, we indicate the available variables. MIMIC stands for MIMIC-III. *(continued)*

| Name | Description | MIMIC | eICU | HiRID | AUMC | Emory |
|---|---|---|---|---|---|---|
| cai | calcium ionized | ✓ | ✓ | ✓ | ✓ | ✗ |
| ck | creatine kinase | ✓ | ✓ | ✓ | ✓ | ✗ |
| ckmb | creatine kinase MB | ✓ | ✓ | ✓ | ✓ | ✗ |
| cl | chloride | ✓ | ✓ | ✓ | ✓ | ✓ |
| crea | creatinine | ✓ | ✓ | ✓ | ✓ | ✓ |
| crp | C-reactive protein | ✓ | ✓ | ✓ | ✓ | ✗ |
| dbp | diastolic blood pressure | ✓ | ✓ | ✓ | ✓ | ✓ |
| eos | eosinophils | ✓ | ✓ | ✗ | ✓ | ✗ |
| esr | erythrocyte sedimentation rate | ✓ | ✗ | ✓ | ✓ | ✗ |
| etco2 | endtidal CO2 | ✓ | ✗ | ✓ | ✓ | ✓ |
| fgn | fibrinogen | ✓ | ✓ | ✓ | ✓ | ✓ |
| fio2 | fraction of inspired oxygen | ✓ | ✓ | ✓ | ✓ | ✓ |
| glu | glucose | ✓ | ✓ | ✓ | ✓ | ✓ |
| hbco | carboxyhemoglobin | ✗ | ✓ | ✓ | ✓ | ✗ |
| hct | hematocrit | ✓ | ✓ | ✗ | ✓ | ✓ |
| height | patient height | ✓ | ✓ | ✓ | ✓ | ✗ |
| hgb | hemoglobin | ✓ | ✓ | ✓ | ✓ | ✓ |
| hr | heart rate | ✓ | ✓ | ✓ | ✓ | ✓ |
| inr_pt | prothrombin time/international normalized ratio | ✓ | ✓ | ✓ | ✓ | ✗ |
| k | potassium | ✓ | ✓ | ✓ | ✓ | ✓ |
| lact | lactate | ✓ | ✓ | ✓ | ✓ | ✓ |
| lymph | lymphocytes | ✓ | ✓ | ✓ | ✓ | ✗ |
| map | mean arterial pressure | ✓ | ✓ | ✓ | ✓ | ✓ |
| mch | mean cell hemoglobin | ✓ | ✓ | ✓ | ✓ | ✗ |
| mchc | mean corpuscular hemoglobin concentration | ✓ | ✓ | ✓ | ✓ | ✗ |
| mcv | mean corpuscular volume | ✓ | ✓ | ✓ | ✓ | ✗ |
| methb | methemoglobin | ✓ | ✓ | ✓ | ✓ | ✗ |
| mg | magnesium | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 4.1: Variables provided to the models for predicting sepsis. For each dataset, we indicate the available variables. MIMIC stands for MIMIC-III. *(continued)*

| Name | Description | MIMIC | eICU | HiRID | AUMC | Emory |
|---|---|---|---|---|---|---|
| na | sodium | ✓ | ✓ | ✓ | ✓ | ✗ |
| neut | neutrophils | ✓ | ✓ | ✓ | ✓ | ✗ |
| o2sat | oxygen saturation | ✓ | ✓ | ✓ | ✓ | ✓ |
| pco2 | CO2 partial pressure | ✓ | ✓ | ✓ | ✓ | ✓ |
| ph | pH of blood | ✓ | ✓ | ✓ | ✓ | ✓ |
| phos | phosphate | ✓ | ✓ | ✓ | ✓ | ✓ |
| plt | platelet count | ✓ | ✓ | ✓ | ✓ | ✓ |
| po2 | O2 partial pressure | ✓ | ✓ | ✓ | ✓ | ✗ |
| pt | prothrombine time | ✓ | ✓ | ✗ | ✓ | ✗ |
| ptt | partial thromboplastin time | ✓ | ✓ | ✓ | ✓ | ✓ |
| rbc | red blood cell count | ✓ | ✓ | ✗ | ✓ | ✗ |
| rdw | erythrocyte distribution width | ✓ | ✓ | ✗ | ✓ | ✗ |
| resp | respiratory rate | ✓ | ✓ | ✓ | ✓ | ✓ |
| sbp | systolic blood pressure | ✓ | ✓ | ✓ | ✓ | ✓ |
| sex | patient sex | ✓ | ✓ | ✓ | ✓ | ✓ |
| tco2 | totcal CO2 | ✓ | ✓ | ✗ | ✗ | ✗ |
| temp | temperature | ✓ | ✓ | ✓ | ✓ | ✓ |
| tnt | troponin t | ✓ | ✓ | ✓ | ✓ | ✗ |
| tri | troponin I | ✓ | ✓ | ✗ | ✗ | ✓ |
| urine | urine output | ✓ | ✓ | ✓ | ✓ | ✗ |
| wbc | white blood cell count | ✓ | ✓ | ✓ | ✓ | ✓ |
| weight | patient weight | ✓ | ✓ | ✓ | ✓ | ✗ |

### 4.2.3 PREDICTION METHODS

Next, we give an overview of the prediction methods included in this study. Specifically, for machine learning methods we considered i) a deep self-attention model (attn) [201], ii) a recurrent neural network with gated recurrent units (gru), iii) a light gradient boosting machine (lgbm) [104], and iv) a LASSO-regularised [197] logistic regression (lr) model. These ML methods can be grouped into deep learning (DL) methods (attn and gru) and classical ML methods rooted in statistical learning (lgbm and lr). Furthermore, we included several clinical scores as baselines to test their usefulness for predicting sepsis. This involves v) sequential organ failure assessment (SOFA) [204], vi) quick SOFA (qSOFA) [185], vii) systemic inflammation response syndrome (SIRS) [17], viii) National Early Warning Score (NEWS) [100], and ix) Modified Early Warning Score (MEWS) [190].

All ML models were trained to minimise the binary cross-entropy (BCE) loss between the predicted score and the binary prediction target (which was defined in Section 4.2.2) computed over all hourly time steps. As the non-DL models were cheap to fit (in terms of required compute), we ran a randomised search over 50 iterations (each corresponding to a hyperparameter configuration) of a 5-fold cross-validation (CV). The CV was stratified for sepsis cases, i.e., preserving the class imbalance in all folds, while we additionally ensured that subsequent observations from the same ICU stay were not subdivided across folds. For the DL methods, hyperparameters such as the learning rate, width, depth, batch size, weight decay, dropout, as well as model checkpoints were selected by minimising the BCE loss on a hold-out set which was created on-the-fly by withholding 10% of the samples from the training set. We refer to this loss as our "online validation loss".

All DL models were trained for at most 100 epochs, where we stopped training early if the online validation loss did not improve for 20 epochs. Instead of an out-of-the-box 50 iterations of randomised search, since the DL models were a bit more delicate to properly tune, we divided this contingent of 50 hyperparameter configurations to be tested into two parts: In a first *coarse* search, we evaluated 25 random configurations of a pre-defined hyperparameter grid (see Table 4.2). Then, we configured tighter ranges surrounding the best performing configuration of the coarse search to run a *fine* hyperparameter search, which evaluated another 25 hyperparameter configurations. For the second step, we kept the architecture fixed (depth, width and batch size) in order to fine-tune the regularisation, in terms of weight decay and dropout, as well as the learning rate. Even though regularisation was the main focus of the fine tuning step, we also applied dropout and weight decay in the coarse search to prevent a scenario where heavily parametrised models (that after fine-tuning could perform very well) are discarded in the coarse search step due to overfitting upon being trained without any regularisation.

For all ML models, we employed class weights inversely proportional to the prevalence of the positive class (on the time step-level). This enabled a loss function, which overall assigns the same weighting to the few positively labelled observations as it does to the many negatively labelled ones. All hyperparameter searches were performed on the training set of the first repetition of splitting the derivation set (see Section 4.2.5) into training and validation data. After tuning the hyperparameters, the best hyperparameter configuration was reused in all five training sets (corresponding to the five repetitions of splitting) in order

| Hyperparameter | Coarse search values |
|---|---|
| Depth | gru: $1, 2, 3$, attn: $2$ |
| Width | $32, 64, 128, 256$ |
| Learning rate | log uniformly in range $e^{-9}$ – $e^{-7}$ |
| Dropout | $0.3, 0.4, 0.5, 0.6, 0.7$ |
| Weight decay | $0.0001, 0.001, 0.01, 0.1$ |

Table 4.2: Hyperparameter grid and ranges for the coarse hyperparameter search of the recurrent neural network with gated recurrent units (gru) and the self-attention model (attn).

to fit five repetition models to assess performance robustness under varying training data. Each repetition model was then applied to the test split for evaluating and reporting the final performance metrics. This procedure was applied on each dataset independently.

Next, we provide more specifications about the four ML model architectures, starting with the self-attention model.

SELF-ATTENTION MODEL For the attention model, we apply positional encoding (PE) to the relative observation times that are measured as the number of hours since ICU admission. Each time step $t$ is mapped to a 10-dimensional encoding such that

$$\mathrm{PE}(t, 2i) = \sin(t \cdot s_i), \text{ and} \tag{4.1}$$

$$\mathrm{PE}(t, 2i+1) = \cos(t \cdot s_i), \tag{4.2}$$

where $i \in \{0, \ldots, 4\}$ enumerates the five different time scales that are used. The actual time scalings $s_i$ were computed as

$$s_i = a \cdot \exp\left(\frac{-\log\left(\frac{b}{a}\right) \cdot i}{S - 1}\right), \tag{4.3}$$

where $a, b$ represent the minimal and maximal time scales and $S$ denotes the number of employed time scales. Here, we used $a = 1, b = 500$, and $S = 5$. The ten dimensions of positional embedding are then concatenated with the 59 dimensions of time series data. An initial linear layer maps the above sequence to a sequence of model dimension $d$. Next, we sequentially apply two Transformer layers, where each layer contains the following sequence of transformations:

1. a multi-head attention layer using causal masking to prevent future data leakage,

2. a residual connection adding the above output to the input of the attention layer,
3. a multilayer perceptron (MLP) using one hidden dimension of size $4d$ and rectified linear unit (ReLU) activations, and
4. another residual mapping to combine the outputs of step 2 and 3.

Finally, after the causally-masked Transformer layers (or sequential Transformer layers), a final linear layer maps the $d$-dimensional representation to a one-dimensional output, which we treat as the logits of the binary classification problem.

GRU MODEL For our recurrent neural network, we provide the 59 time series variables as input to the recurrent architecture. Furthermore, we used the four static variables to initialise the model state (via a linear projection to the model dimension). To allow for a comparable number of parameters to the attn model, besides the width (or model dimension,) we also varied the number of layers to allow for a depth up to three layers (see Table 4.2).

LGBM MODEL For the LightGBM model, we used the following grid of hyperparameters of which $50$ random configurations were then evaluated during the hyperparameter search. For the number of estimators, we allowed the values $100, 300, 500, 1{,}000, 2{,}000$. Further hyperparameters include the boosting type (“gbdt” or “dart”), the learning rate $(0.001, 0.01, 0.1, 0.5)$, the number of leaves used $(30, 50, 100)$, and finally the $L_1$ regularisation strength $(0, 0.1, 0.5, 1, 3, 5)$.

LR MODEL For training the logistic regression model, we considered two alternative optimisers: “saga” and “liblinear”. For the LASSO penalty ($L_1$ regularisation), we log-uniformly partitioned the range $(10^{-3}, 10^{2})$ into $50$ values to choose from in the hyperparameter search.

### 4.2.4 FEATURE ENGINEERING

Instead of merely providing the current observations to the models, we extracted a diverse set of features that accounts for data missingness and incorporates knowledge about past measurements but also infuses the models with clinical domain knowledge. For this, we accompany the 59 observed time series variables with 59 binary missingness indicators and measurement counts each. Also, we compute 9 clinically derived features that are composed of ratios of variables (such as the shock index) and partial scores (e.g. SOFA), but only including vital and laboratory measurements as part of the set of input variables. Adding the four static variables, the DL models were therefore provided 190 features. In preliminary tests, we observed that incorporating the static variables to the attn model led to a minor decrease in performance (not so with other models such as gru). In a first approach, static variables were either linearly projected to the model dimension to become an additional token preceding the time series tokens in the attention layer. In a second approach, we concatenated the static variables to the time series channels and repeated them accordingly. In both cases, we observed a slight detrimental effect, therefore we disregarded the statics in the attn model. The non-DL models (lr and lgbm) are not directly intended for streams of input data, but rather typically expect fixed-sized vectors. Thus, to make the temporal dynamics governing the patient data available to those two methods, we extracted an additional battery of temporal features from the 59 sequentially observed input variables as well as from the 9 derived features. This includes look-back statistics (median, mean, variance, minimum, and maximum) that were computed over time windows at multiple scales (16, 8, and 4 hours). As these methods were merely provided the feature vectors of individual time steps, to further increase the availability of past information at the current prediction step, in addition to the temporal features, we added a carry-forward imputed channel for each of the 59 time series channels. Thus, adding together all of the above, the classical ML models were provided a rich set of 1,269 features.

### 4.2.5 Experimental setup

For each included dataset, we define a derivation set (90% of the data) and a hold-out test set (the remaining 10%). In five independent runs, the derivation set of each dataset was split into a training set (80% of the full dataset) that makes up the actual training data, and a validation set (10% of the full dataset) that was used for tuning the models. To preserve the prevalence of sepsis cases in the splits, we applied stratified splitting. All models were optimised on the training and validation split (of the five partitions). After having fixed the model hyperparameters, we fitted model repetitions on all five training splits in order to characterise performance variability under varying training data. Performance metrics were then evaluated by applying these model repetitions to the hold-out test split. To achieve a maximal comparability between internal validations, and external validations (see Section 4.2.6), we quantify the performance metrics on the identical hold-out test splits in both settings. In order to make the performance metrics comparable across datasets, we harmonised the sepsis prevalence to the across-dataset mean of 17%. This was implemented by means of subsampling: for increasing the prevalence, we subsampled the controls. Conversely, to reduce the prevalence, we subsampled from the cases. To further ensure that we do not discard large parts of the case cohort, which is of particular interest, we repeated the random subsampling process ten times and confirmed that the vast majority of sepsis cases (over 98.3%) were covered in those sets that were then used in the evaluation.

### 4.2.6 Evaluation

Patient-focused evaluation strategy In our analysis, we consider an online prediction scenario, where patients are continuously monitored and predictions are being made in hourly intervals. However, even though the models employed in this analysis consider patients on a *time point*-level, in order to arrive at a clinically meaningful evaluation strategy, we consider performance metrics on the *patient*-level. This includes:

i) area under the receiver-operating-characteristic curve (AUROC), and

ii) positive predictive value (PPV) and alarm earliness at a prediction threshold fixed at 80% sensitivity.

First, we use the unnormalised prediction scores (logits) and, for sake of robustness, consider the innermost 99 percentiles[4]. Then, we partition the remaining range of scores into 100 evenly-spaced thresholds. For each threshold, we then swept through the ICU stay and as soon as the models prediction score surpasses the current threshold, we trigger an alarm for sepsis and register the alarm time. After having triggered a single alarm for a given threshold, the alarm system is stopped, and we evaluate the next threshold. Motivated by its clinical use case, this evaluation strategy is not exhaustive in the sense that no second alarms are considered after sepsis has been suspected by the alarm system. This is in line with previous work considering clinically useful evaluation strategies of prediction models [208]. Next, the set of alarms for a given threshold were used to fill a corresponding confusion matrix, where having raised an alarm for a sepsis case is considered a true positive, raising an alarm in a control patient would be counted as a false positive, and not raising an alarm in a control would count as true negative, and so on. Receiver operating characteristic (ROC) curves were then computed using the entries of the confusion matrices, over all thresholds.

For the second evaluation metric, again making use of the confusion matrices, we first determine the threshold (and the corresponding confusion matrix) that leads to a sensitivity of 80%. For the identified threshold, we then reported PPV (or precision) as well as alarm earliness which was defined as the median number of hours that the alarm was raised ahead of sepsis onset. We deliberately chose the median as the more robust summary statistic, since a few very early alarms could otherwise lead to overly optimistic results and interpretations,

---

[4] This step is done merely for defining thresholds in the evaluation, so we do not actually discard predictions that were made.

e.g. when reporting mean earliness. If no threshold exactly coincided with 80% sensitivity, we instead used the two closest thresholds (above and below the target sensitivity) and linearly interpolated the sought-after performance measures. Stopping the alarm evaluation after the first raised alarm leads to a conservative evaluation in that no repeated alarms are admissible. While repeated alarms would improve sensitivity for sepsis, it would also come at the heavy cost of false positives that could lead to alarm fatigue. While this scenario makes the prediction task more challenging, it also upper bounds the number of false alarms in a control stay to at most 1.

We reported mean and standard deviation (SD) for all performance measures by first computing the mean over the ten subsamplings and then reporting the mean and SD over the five repetition splits. Additionally, we computed 95% confidence intervals (CI) by considering all 50 iterations (5 repetition splits and 10 subsamplings) as bootstrap samples in order to calculate percentile intervals.

INTERNAL AND EXTERNAL VALIDATION As a first step, for a given method we trained a separate model on the derivation set of each dataset. This was repeated for all investigated methods. Next, as an internal validation, we evaluated the performance metrics on the hold-out test set of the same dataset that the model was trained on, respectively. As for an external validation, we considered a given dataset and method and pool all models that were trained on the remaining datasets by choosing the maximal predicted score at each hourly prediction step. We refer to this scenario as *pooled* predictions. Additionally, we also report the performance that is observed when applying a model that was trained on one dataset and then applied to another one, and refer to this scenario as *pair-wise* predictions, where "pair" refers to the two considered datasets.

CALIBRATION Model calibration was assessed by means of reliability diagrams that were computed on the hold-out test splits. In order to calibrate the models, we used Platt scaling as tuned on the respective validation split. To conform with the analysis regarding model discrimination, we also consider predictions on the patient-level for model calibration. The reliability diagrams are computed using the patient-level sepsis labels to derive the fraction of positives (or true risk) in a given bin of ICU stays. Furthermore, we considered the average predicted score of a stay as the predicted risk, in order to estimate the mean predicted risk over a bin of ICU stays.

VARIABLE IMPORTANCE Finally, we determined the relevance and contributions of individual variables to the overall predictions via the calculation of Shapley values [129] using the integrated gradients method [191]. For this, we considered the raw measurements of laboratory and vital measurements (as opposed to other extracted features such as counts or missingness indicators) in order to potentially retrieve pathophysiological signals that are indicative of sepsis. To reduce the memory footprint of the integrated gradients method, we randomly sampled 500 ICU stays from the hold-out test set of each dataset and repeated this process five times. For each stay, we considered the time step with the maximal predicted score as the potentially most interesting anchor point, i.e., the point of time where the model is most suspicious about an imminent sepsis. Then, for each stay we assessed how the model is affected by changes in the value of individual observations and channels up to 16 hours preceding these anchor points, resulting in a Shapley value for each variable and for up to 16 hourly time steps ahead of the maximal predicted score. We designed this procedure in this way in order to focus on interesting time windows amidst potentially long ICU stays that could swamp this analysis with noise. For the same reason, we considered only recent time steps ahead of the maximal predicted score to ensure that we attend to time steps that plausibly contain interesting signals, as opposed to unrelated time points that could stem from up to one week in the past.

## 4.3 RESULTS

We first present a description of the investigated multi-centre ICU cohort in Section 4.3.1. In the remaining sections, we report the empirical results of all included methods, with a particular focus on the best performing (and also most flexible) model: the deep self-attention model (attn).

### 4.3.1 DATASET CHARACTERISTICS

After preprocessing and filtering, our harmonised cohort consisted of 156,309 unique ICU stays that correspond to over 783 years worth of ICU data. Out of these ICU stays, 26,734 (17.1%) developed sepsis (as defined by Sepsis-3). In Table 4.3, we characterise our ICU cohort in terms of summary statistics. Four of the used datasets, MIMIC-III, eICU, HiRID, and AUMC, were processed in this study and showed a large overlap in the availability of a core set of 63 variables. Therefore, we subsequently refer to them as the *core* datasets. The remaining dataset, originating from the Emory hospital, showed a smaller overlap to the core set of variables, reporting only 35 out of the core set of 63 variables. Due to this, and also since the Emory dataset was the only dataset that was provided with precomputed sepsis labels (that could not be validated due to the non-availability of the necessary data), we report analyses from this dataset separately (see Section A.1.1).

| Variable | MIMIC-III | eICU | HiRID | AUMC | Emory |
|---|---|---|---|---|---|
| Cohort size (n) | 36,591 | 56,765 | 27,278 | 15,844 | 19,831 |
| Sepsis-3 prevalence (n (%)) | 9,541 (26) | 4,708 (8) | 10,170 (37) | 1,275 (8) | 1,050 (5) |
| Age, years (Median (IQR)) | 65 (52-77) | 65 (53-76) | 65 (55-75) | 65 (55-75) | 62 (50-72) |
| Ethnicity (%) | | | | | |
| African American | 9 | 10 | - | - | - |
| Asian | 2 | 1 | - | - | - |
| Caucasian | 71 | 82 | - | - | - |
| Hispanic | 3 | 2 | - | - | - |
| Other | 15 | 5 | - | - | - |
| In-hospital mortality (%) | 8 | 7 | 5 | 5 | - |
| ICU LOS, days (Median (IQR)) | 1.99 (1.15-3.63) | 1.71 (0.95-3.01) | 0.97 (0.8-1.95) | 0.97 (0.81-1.82) | - |
| Hospital LOS, days (Median (IQR)) | 6.43 (3.82-11.14) | 5.53 (2.99-9.89) | - | - | - |
| Gender, female (%) | 44 | 45 | 37 | 35 | 46 |
| Gender, male (%) | 56 | 55 | 63 | 65 | 54 |
| Ventilated patients (n (%)) | 16,499 (45) | 24,534 (43) | 14,021 (51) | 10,469 (66) | - |
| Patients on vasopressors (n (%)) | 9,669 (26) | 6,769 (12) | 7,721 (28) | 7,980 (50) | - |
| Patients on antibiotics (n (%)) | 21,598 (59) | 21,847 (38) | 17,152 (63) | 11,165 (70) | - |
| Patients with suspected infection (n (%)) | 16,349 (45) | 9,739 (17) | 15,160 (56) | 1,639 (10) | - |
| Initial SOFA (Median (IQR)) | 3 (1-4) | 3 (1-5) | 5 (3-8) | 6 (3-7) | - |
| SOFA components (Median (IQR)) | | | | | |
| Respiratory | 1 (0-2) | 1 (0-2) | 3 (2-4) | 2 (1-3) | - |
| Coagulation | 0 (0-1) | 0 (0-1) | 0 (0-1) | 0 (0-1) | - |
| Hepatic | 0 (0-1) | 0 (0-0) | 0 (0-1) | 0 (0-0) | - |
| Cardiovascular | 1 (1-1) | 1 (0-1) | 1 (1-4) | 2 (1-4) | - |
| CNS | 0 (0-1) | 0 (0-2) | 0 (0-1) | 0 (0-1) | - |
| Renal | 0 (0-1) | 0 (0-1) | 0 (0-0) | 0 (0-1) | - |
| Admission type (%) | | | | | |
| Surgical | 38 | 19 | - | 80 | - |
| Medical | 61 | 79 | - | 15 | - |
| Other | 1 | 3 | - | 5 | - |

Table 4.3: Dataset description of our ICU cohort. The different datasets vary in the amount of available metadata to characterise the cohorts. While a majority of the data entries are available for the four core datasets that were preprocessed in this study (MIMIC-III, eICU, HiRID, and AUMC), many entries are missing in the Emory dataset, the one dataset that was reused in a preprocessed state [166].

### 4.3.2 INTERNAL VALIDATION

For the internal validation, we assessed the performance of each model on an out-of-sample test set that originates from the same database as the data used for training said model. We display the performance metrics of the internal validation analysis in the left panels of Figures 4.6 to 4.9, respectively. For the best performing method, the self-attention model (attn) model, we observe an average AUROC of 0.847 (95% CI, 0.840 to 0.853) on the dataset-internal hold-out test set, where the average is computed over the four core datasets that share the harmonised set of 63 core variables (see Table 4.1). Furthermore, at a sensitivity of 80%, this model recognised septic patients with a PPV of 39.3% (95% CI, 37.6 to 41.1) and a median alarm earliness of 3.7 hours (95% CI, 3.0 to 4.4) before onset. In other words, this corresponds to raising 1.5 false alarms per true alarm, on average. Notably, we deem the SOFA score as a strong baseline, since it plays a central role in the Sepsis-3 definition, while it also incorporates further information that was not made available to the ML models such as vasopressor administrations (and dosages) or neurological examinations as summarised in the GCS score. Nevertheless, on all datasets, we observe substantial improvements in terms of AUROC when comparing the deep self-attention model with this baseline. Finally, in Figure A.7 we display an auxiliary analysis where we pooled the actual data (as opposed to only pooling the prediction scores) such that for a given testing dataset the data of the remaining datasets were pooled for *training*. Interestingly, we did not observe improved performance when pooling the data as compared to our federated learning setting using pooled predictions, where instead of sensitive patient data only trained prediction models need to be shared across the centers.

### 4.3.3 EXTERNAL VALIDATION

In our external validation, we apply previously trained models to independent databases for testing. Results regarding *pair-wise* predictions, i.e., training on one dataset and predicting on another one, are shown in Figure 4.10 for the deep self-attention model (attn) and in Figures A.8 to A.17 of the Supplementary Materials in more detail for all methods. Furthermore, the external validation performance of *pooled* predictions is shown in the last row of the heat map in Figure 4.10, whereas pooled predictions for all methods are shown in the right panels of Figures 4.6 to 4.9, respectively. Using the proposed pooling strategy in our external validation, our deep self-attention model on average achieves an AUROC of 0.76 (95% CI, 0.747 to 0.770). Fixing the prediction threshold at a sensitivity of 80%, we observe a PPV of 29.3% (95% CI, 28.0 to 30.9) at a median earliness (lead time to onset) of 1.75 hours (95% CI, 0.88 to 2.81). Figure 4.10 suggests that applying the pooling strategy to a given dataset

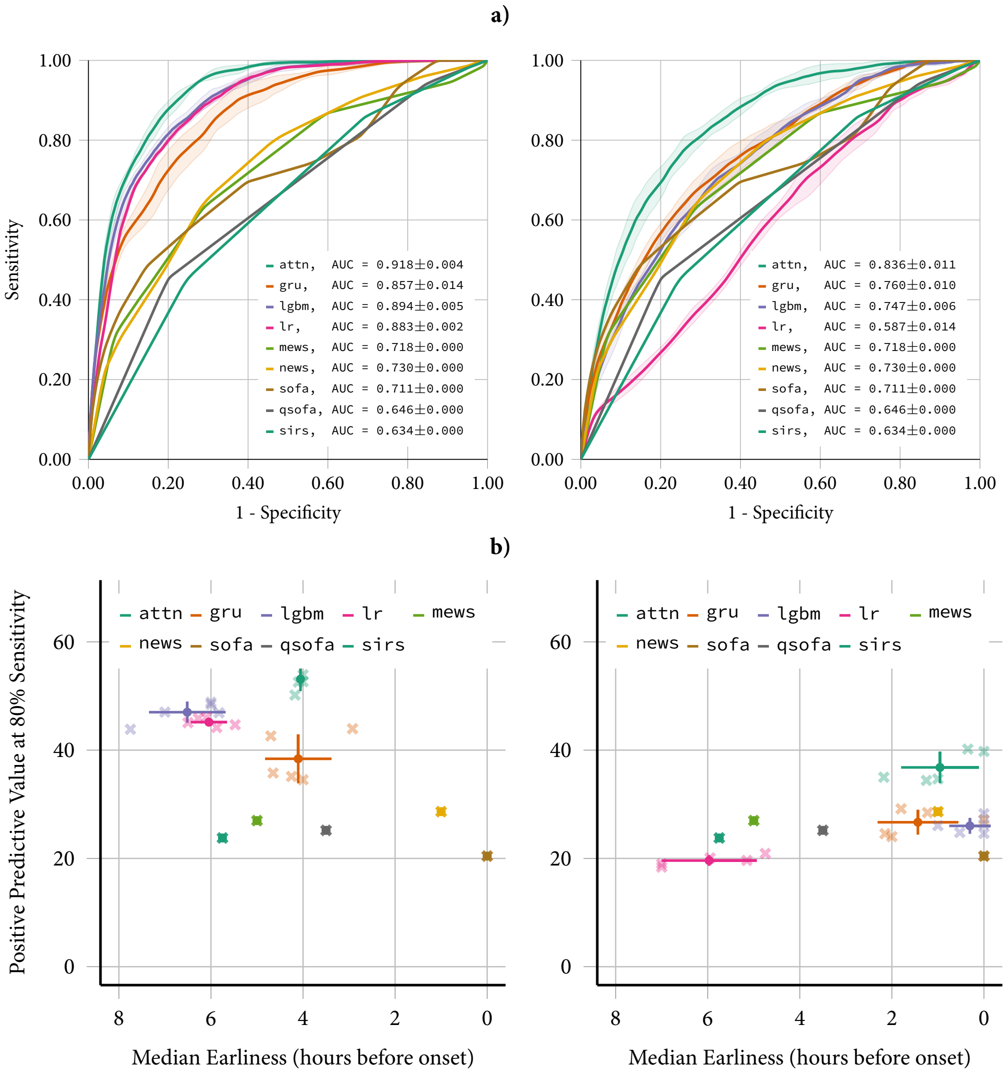


Figure 4.6: Performance plots for the AUMC dataset and all considered methods. The left panels display the internal validation performance, evaluated on the hold-out test set of the database that was used for training. The right panels show the external validation performance using *pooled* predictions (see Section 4.2.6). In row **a)**, ROC curves are shown, whereas in row **b)** positive predictive value (PPV) and alarm earliness are shown when fixing sensitivity at 80%. We report the following ML approaches: a deep self-attention model (attn), a recurrent neural network using gated recurrent units (gru), a light gradient boosting machine (lgbm), and a logistic regression (lr) model. Furthermore, we display the following clinical scores: Modified Early Warning Score (MEWS), National Early Warning Score (NEWS), sequential organ failure assessment (SOFA), quick SOFA (qSOFA), and systemic inflammation response syndrome (SIRS). For the ML models, we show the mean ± standard deviation (SD) over the five repetitions of splitting the derivation set, whereas the clinical scores were not affected by varying training data (hence no variation). In row **a)** means are shown as lines, SDs as faded bands; in row **b)** results for the individual split repetitions are shown as tilted and faded crosses, whereas SDs are shown as solid horizontal and vertical lines surrounding the means (round solid dots).

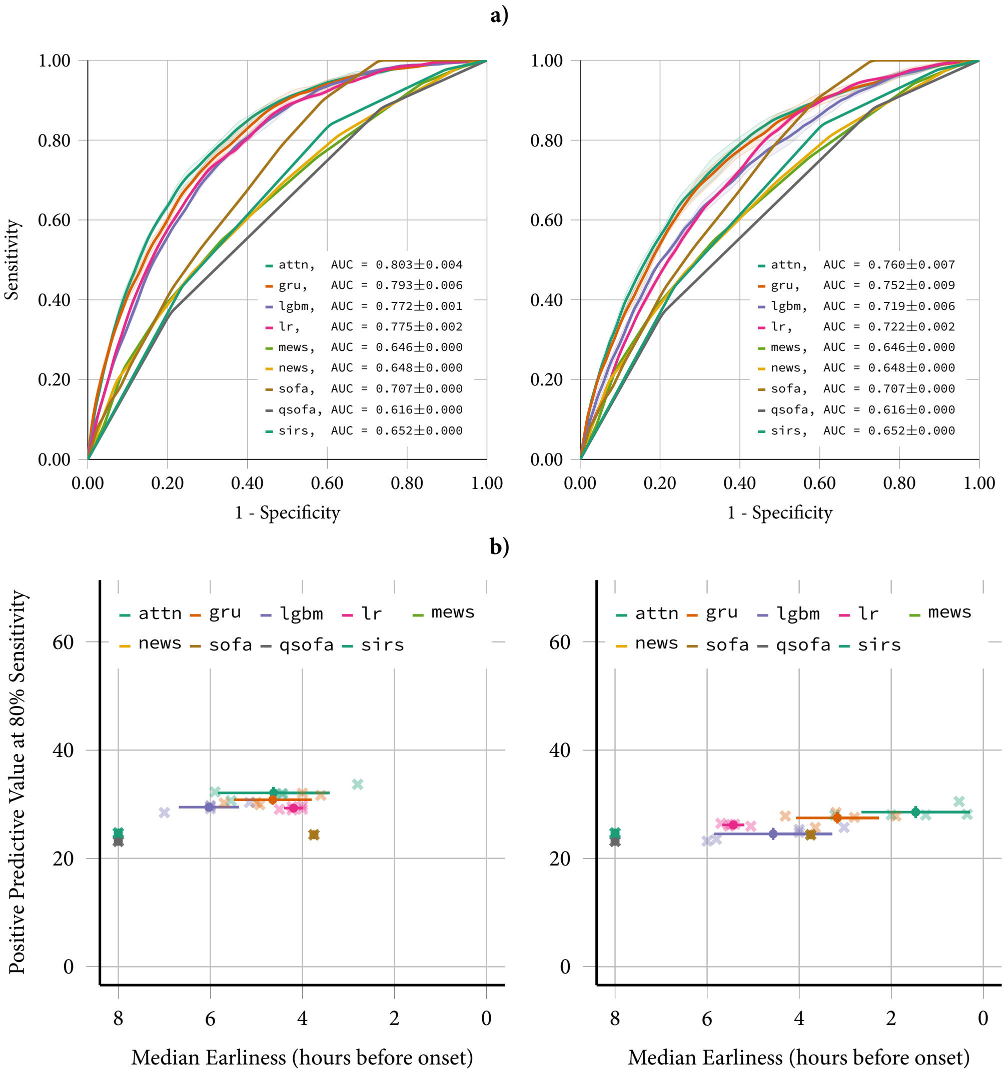


Figure 4.7: Performance plots for the eICU dataset and all considered methods. The left panels display the internal validation performance, evaluated on the hold-out test set of the database that was used for training. The right panels show the external validation performance using *pooled* predictions (see Section 4.2.6). In row **a)**, ROC curves are shown, whereas in row **b)** positive predictive value (PPV) and alarm earliness are shown when fixing sensitivity at 80%. We report the following ML approaches: a deep self-attention model (attn), a recurrent neural network using gated recurrent units (gru), a light gradient boosting machine (lgbm), and a logistic regression (lr) model. Furthermore, we display the following clinical scores: Modified Early Warning Score (MEWS), National Early Warning Score (NEWS), sequential organ failure assessment (SOFA), quick SOFA (qSOFA), and systemic inflammation response syndrome (SIRS). For the ML models, we show the mean ± standard deviation (SD) over the five repetitions of splitting the derivation set, whereas the clinical scores were not affected by varying training data (hence no variation). In row **a)** means are shown as lines, SDs as faded bands; in row **b)** results for the individual split repetitions are shown as tilted and faded crosses, whereas SDs are shown as solid horizontal and vertical lines surrounding the means (round solid dots).

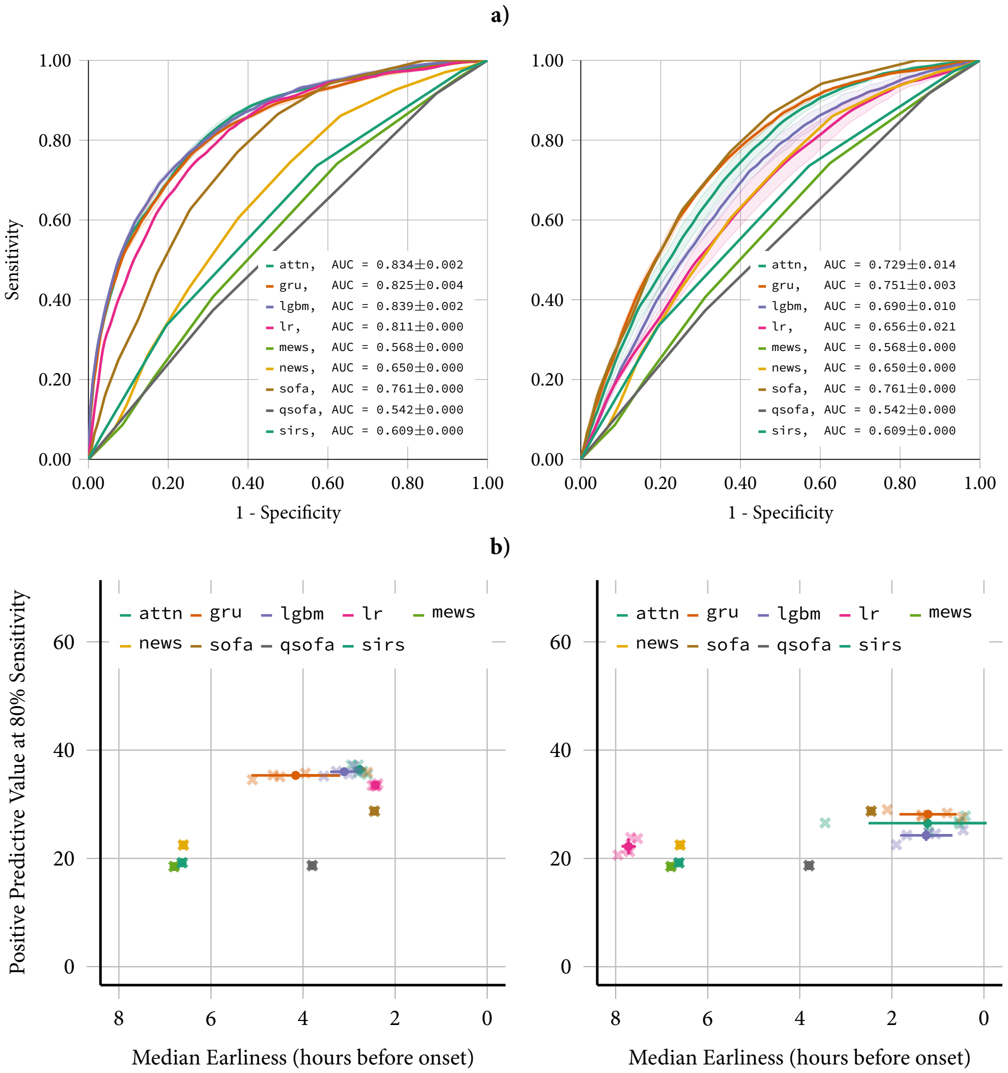


Figure 4.8: Performance plots for the HiRID dataset and all considered methods. The left panels display the internal validation performance, evaluated on the hold-out test set of the database that was used for training. The right panels show the external validation performance using *pooled* predictions (see Section 4.2.6). In row **a)**, ROC curves are shown, whereas in row **b)** positive predictive value (PPV) and alarm earliness are shown when fixing sensitivity at 80%. We report the following ML approaches: a deep self-attention model (attn), a recurrent neural network using gated recurrent units (gru), a light gradient boosting machine (lgbm), and a logistic regression (lr) model. Furthermore, we display the following clinical scores: Modified Early Warning Score (MEWS), National Early Warning Score (NEWS), sequential organ failure assessment (SOFA), quick SOFA (qSOFA), and systemic inflammation response syndrome (SIRS). For the ML models, we show the mean ± standard deviation (SD) over the five repetitions of splitting the derivation set, whereas the clinical scores were not affected by varying training data (hence no variation). In row **a)** means are shown as lines, SDs as faded bands; in row **b)** results for the individual split repetitions are shown as tilted and faded crosses, whereas SDs are shown as solid horizontal and vertical lines surrounding the means (round solid dots).

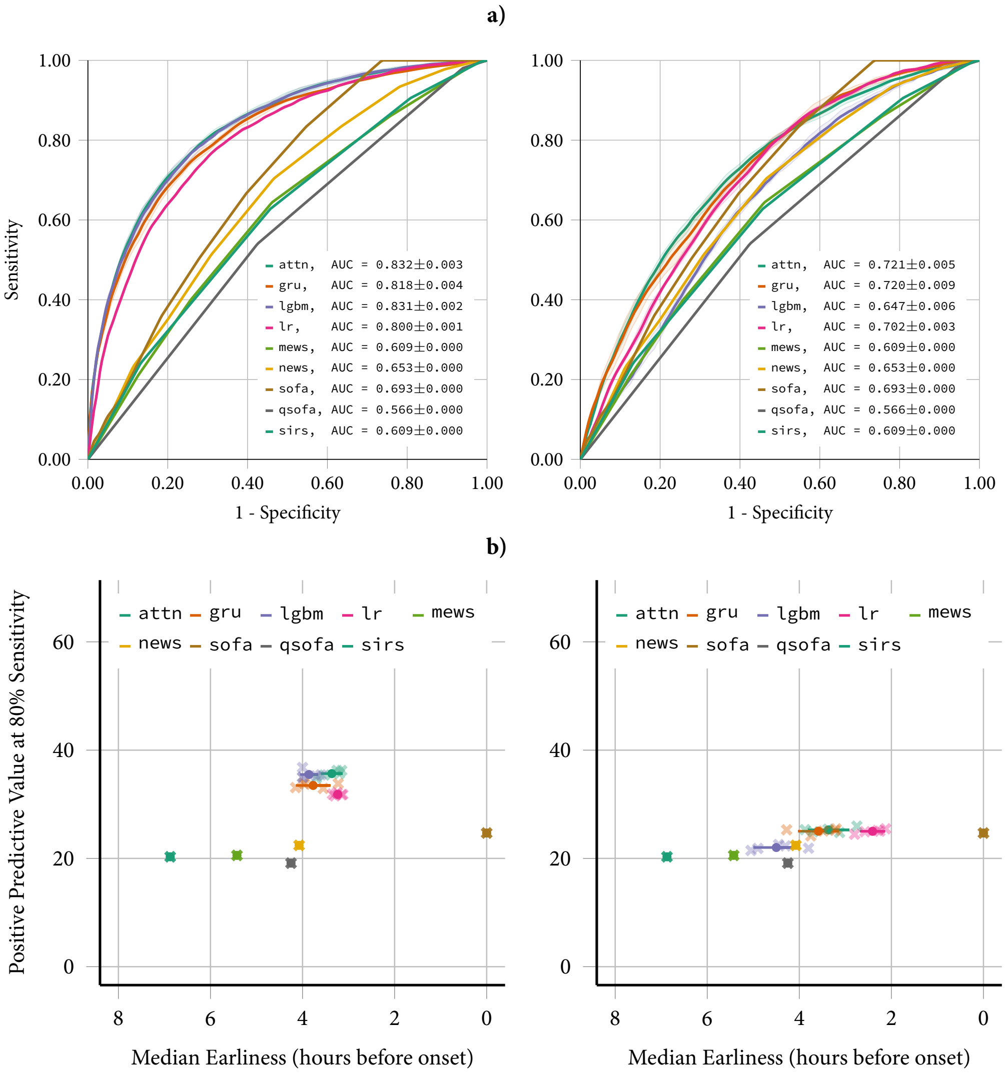


Figure 4.9: Performance plots for the MIMIC-III dataset and all considered methods. The left panels display the internal validation performance, evaluated on the hold-out test set of the database that was used for training. The right panels show the external validation performance using *pooled* predictions (see Section 4.2.6). In row **a)**, ROC curves are shown, whereas in row **b)** positive predictive value (PPV) and alarm earliness are shown when fixing sensitivity at 80%. We report the following ML approaches: a deep self-attention model (attn), a recurrent neural network using gated recurrent units (gru), a light gradient boosting machine (lgbm), and a logistic regression (lr) model. Furthermore, we display the following clinical scores: Modified Early Warning Score (MEWS), National Early Warning Score (NEWS), sequential organ failure assessment (SOFA), quick SOFA (qSOFA), and systemic inflammation response syndrome (SIRS). For the ML models, we show the mean ± standard deviation (SD) over the five repetitions of splitting the derivation set, whereas the clinical scores were not affected by varying training data (hence no variation). In row **a)** means are shown as lines, SDs as faded bands; in row **b)** results for the individual split repetitions are shown as tilted and faded crosses, whereas SDs are shown as solid horizontal and vertical lines surrounding the means (round solid dots).

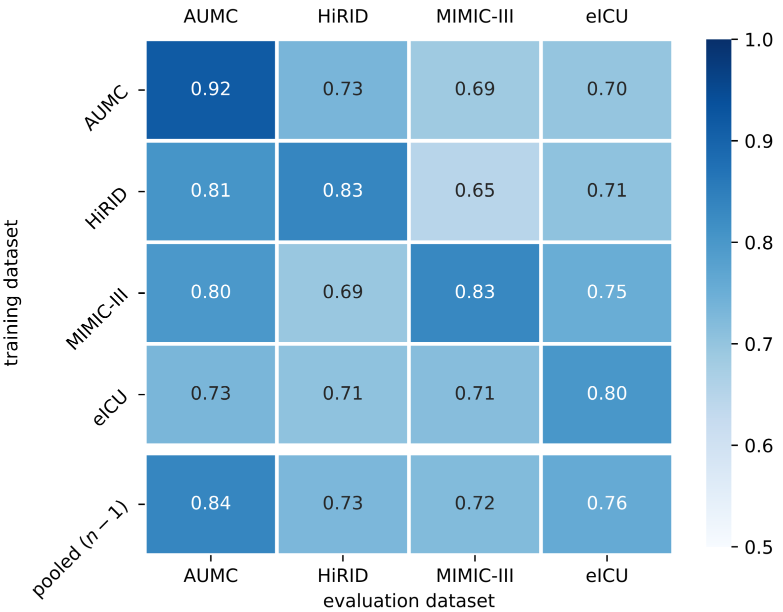


Figure 4.10: External validation results illustrated for our deep self-attention model (attn). Rows indicate the dataset used for training, columns indicate the testing dataset. The cell entries represent the area under the receiver-operating-characteristic curve (AUROC) that was measured on the hold-out test set. The bottom row displays the performance of the *pooled* predictions where for a given testing dataset the predictions from models individually trained on the remaining datasets were aggregated by taking the maximal predicted score. In the first four rows, on-diagonal entries correspond to internal validation performance; off-diagonal entries refer to *pair-wise* predictions, i.e., the pair-wise transfer of a model trained on one dataset and tested on another one.

achieves better or on-par performance when compared to the single best performing model that was trained on one of the remaining datasets and which could only be determined ex post hoc (i.e., only after evaluating all models individually on the dataset of interest).

### 4.3.4 Model inspection

After having presented the internal and external validation results, next we further investigate and inspect the deep self-attention model. First, we aim to explain the model's predictions using feature attributions. Second, we present ablations where we evaluate predictive performance on subcohorts as well as on different feature categories.

Variable importance To assess the relevance and the impact that individual variables had on the attention model's predictions, we present a Shapley analysis in Figure 4.11. Mean absolute Shapley values are shown averaged over all datasets in Figure 4.11a, where we ob-

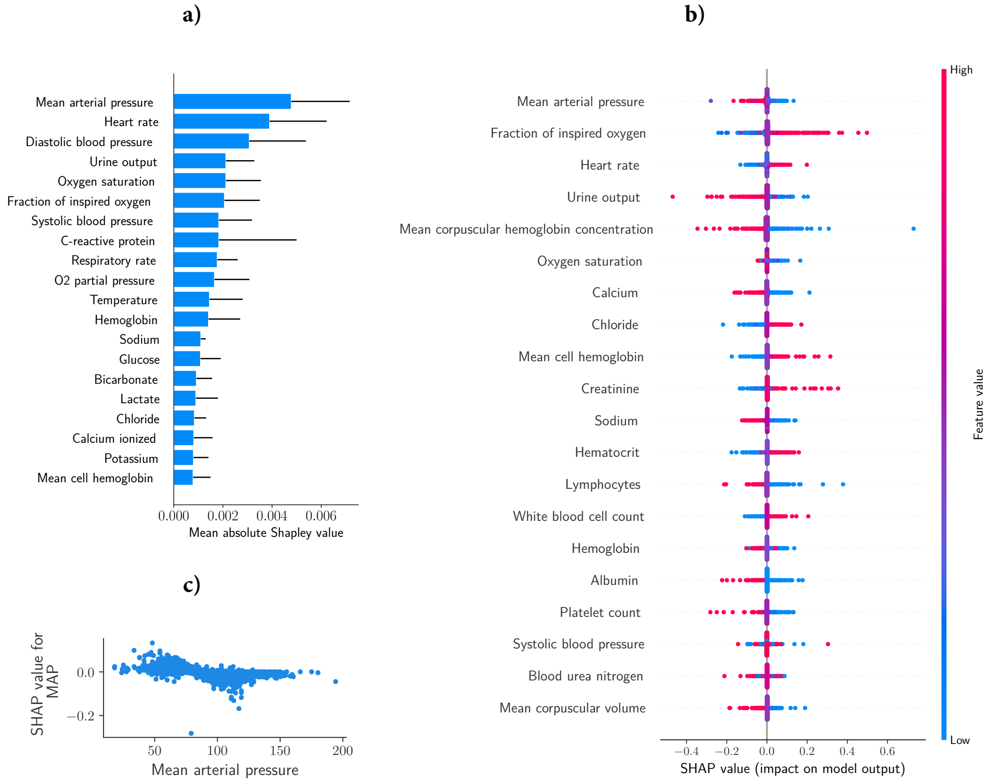


Figure 4.11: Variable importance using Shapley values. In Panel **a)**, we display the mean absolute Shapley values as computed and averaged over all datasets. Error bars indicate the standard deviation across datasets. Here, we show the 20 variables with the largest mean absolute Shapley value, where larger values represent larger contributions to the prediction of sepsis. In Panel **b)**, distributions of Shapley values are illustrated for the eICU dataset. For that, the features were sorted in descending order according to their contribution to the model's predictions. Positive values on the $x$-axis indicate an *increase* in the model's predicted score, i.e., the predicted risk of sepsis, and vice versa for negative values. Additionally, the points are colored to indicate high (red) or low (blue) feature values. For instance, low values in mean arterial pressure is associated with an increase in the model's predicted risk of sepsis. In Panel **c)**, we show the distribution of Shapley values that were computed for the mean arterial pressure variable on the eICU dataset.

serve that the variables mean arterial pressure and heart rate contribute most to the model's predictions. Interestingly, this aligns well with clinical domain knowledge, since clinicians frequently assess these parameters jointly in order to monitor hemodynamic stability, a key determinant of prognosis in ICU patients. Additionally, in Figures 4.11b and 4.11c, for the eICU dataset we display Shapley value distributions that explain whether high or low values of a given variable increase (large Shapley values) or decrease (small Shapley values) the predicted risk of sepsis. The corresponding plots for all datasets are provided in Figures A.1 to A.4 in the Supplementary Materials. While we observe that the exact ranking of variables varies across the different datasets, mean arterial pressure as well as heart rate consistently appear among the top ten variables. Finally, we repeated the Shapley analysis to include all feature types that the attention model uses for prediction (e.g., also measurement counts and missingness indicators) and display the corresponding Shapley values in Figure A.5.

Ablation analyses In Figure 4.12, we investigated whether the foremost surgical composition of the AUMC cohort (as opposed to the other datasets) could be partially responsible for the generally higher performance observed on this dataset. To test this, we reuse the attention models (one for each repetition split) that were trained on the derivation set of the entire AUMC cohort (medical and surgical) and apply them individually to the medical and surgical patients of the hold-out test set of AUMC, i.e., in-distribution data, and to the test set of MIMIC-III, i.e., out-of-distribution data. We found that the surgical cohort of AUMC was indeed easier to classify, however, this pattern did not generalise to the external testing site. Furthermore, given that in Figure A.5 we found that count features frequently appeared in the top ranking variables, we further investigated the performance of models that only use count features or raw observations for prediction, as shown in Figure A.6. While we found no striking difference in performance, we observed a tendency that laboratory measurements (that are less frequently measured) counts are indeed informative, while this was less the case for frequently monitored vital signs.

### 4.3.5 Calibration

Figure 4.13 shows a reliability diagram for the attention models that were trained on the four core datasets without applying any calibration technique. Since the curves skew below the diagonal (which indicates perfect calibration), this suggests that the uncalibrated models tend to be overconfident in that the predicted risk of sepsis (in terms of predicted scores) is higher than the true risk of sepsis (i.e., the sepsis prevalence in a given bin of patients). In Figure 4.14, we display the corresponding reliability diagram upon calibrating the models with Platt scaling. Here, we find that the reliability curves closely track the diagonal, sug-

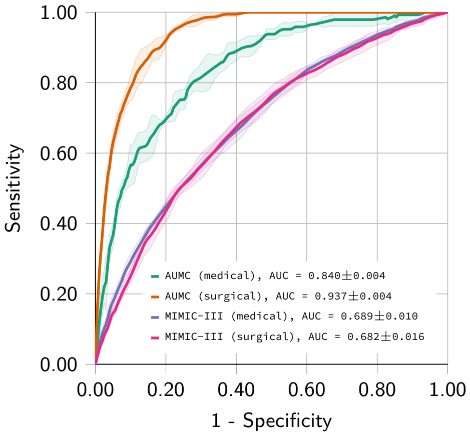


Figure 4.12: ROC plots for the ablation of the subcohorts (medical and surgical patients). For this, attention models trained on AUMC (both admission types) were applied individually to the surgical and medical cohort of the hold-out test set of AUMC (in-distribution) and MIMIC-III (out-of-distribution). While surgical patients in AUMC indeed seem easier to classify correctly, this finding did not generalise to the external testing site.

gesting that these models can be successfully calibrated on a desired target hospital upon deployment.

## 4.4 Discussion

Key findings In this chapter, we have presented the largest international and harmonised ICU dataset to date. Leveraging EHR records from five publicly available ICU databases gathering patient data from three countries, we developed a sepsis early warning system based on a deep self-attention model and performed an extensive external validation across countries and continents. In the internal validation, we observed excellent predictive performance with 1.5 false alarms at a sensitivity of 80%. In the external validation, we also observe a convincing performance, in particular when pooling predictions from models trained on different databases. These findings suggest that our models are leveraging signals that are generalisable and may be used to predict sepsis in unseen hospitals in countries different from the training site.

Relationship to the literature Sepsis represents a major global burden and a leading cause of mortality in critically ill patients [67]. Given that each hour of delayed recognition and intervention increases mortality [54], there is a strong motivation for the data-driven search for biomarkers and signals that are predictive of sepsis. Even though a variety of studies have tried to address sepsis prediction using machine learning, the majority of studies lack

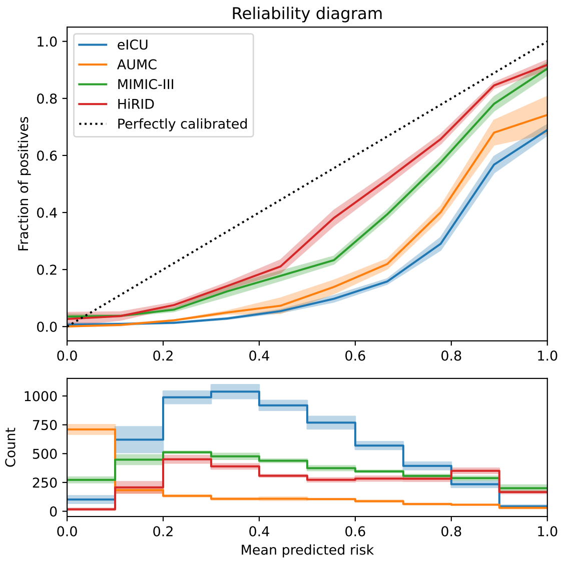


Figure 4.13: Reliability diagram for the attention models trained on the four core datasets *before* calibration, i.e., we display the reliability of the uncalibrated models. Below, a histogram indicates the size of each bin. The reliability curves below the diagonal indicate that the models are overconfident. Figure recreated from Moor et al. [138].

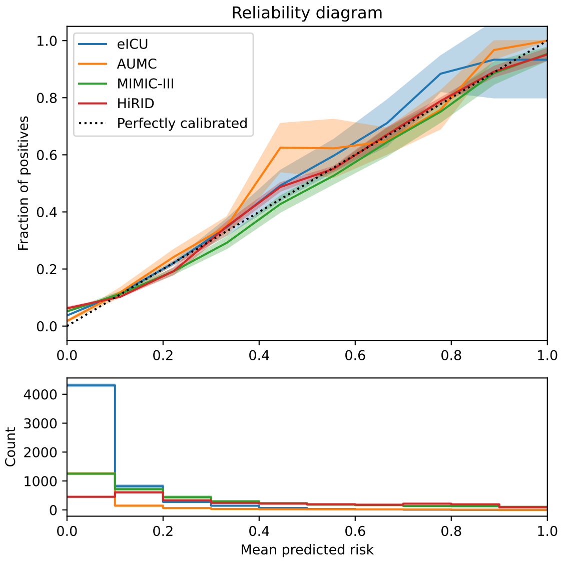


Figure 4.14: Reliability diagram for the attention models trained on the four core datasets after applying Platt scaling to calibrate the models. Below, a histogram indicates the size of each bin. Here, the reliability curves lie close to the diagonal, indicating that they are well calibrated. Figure recreated from Moor et al. [138].

an external validation, and either use the same public dataset, MIMIC-III, or use restricted-access data which further exacerbates the problem of lacking validation data [56, 143]. Most recently, a proprietary tool for sepsis prediction that has been widely adopted across US hospitals performed poorly when externally validated [211], which again emphasises the importance as well as the scale of the problem. By creating a freely available, multi-national ICU dataset with sepsis labels, a key goal of this study was to complement the literature exactly to address the above mentioned challenges, i.e., to enable international external validations, be it before or even after deployment of the models. Furthermore, we showcased a battery of state-of-the-art ML models and found that in particular, a deep self-attention model outperformed its comparison partners as well as clinical baseline scores. Finally, by carrying out an extensive external validation, we observed that this model can generalise to previously unseen hospitals.

The majority of previous studies framed sepsis prediction as an intrinsically retrospective question: Given a sepsis onset at hour $t$, how early could a model have predicted it? This question was then answered by comparing time windows before sepsis onset with time windows in control patients. However, in our systematic review on sepsis prediction [143], we found that a) the particular choice of control windows can drastically affect the performance and the interpretation of the prediction task, and that b) in the majority of studies, this critical detail is *not* reported. This means that high AUROC values may be reported without a guarantee that it would translate to a real-time monitoring scenario. To address this, and to more closely align the retrospective development of the model with a potential deployment scenario, here, we considered an online prediction scenario where both during training and testing, we made repeated predictions in hourly intervals.

IMPLICATIONS OF THE STUDY As one major implication of this study, we showed that it is possible to train deep learning-based sepsis prediction models that can generalise to new hospitals in different countries. Moreover, we found that pooling predictions across datasets can lead to better generalisability when compared to transferring a model from a single dataset to another one. While a single model developed on one dataset may perform poorly on a different dataset due to various types of domain shifts (differences in patient cohorts, ethnicities, treatment policies, monitoring devices, etc.), our findings suggest that these effects are to some degree levelled out when aggregating predictions across different databases. Interestingly, we also observed that our federated learning setup, i.e., combining models that were trained on different cohorts, led to superior performance when compared to pooling of the actual patient data to train a single model on the joint data. This finding is highly encouraging since the pooling of patient data has two significant drawbacks: a) It leads to the costly and repeated retraining of models on much larger datasets. In our external validation, this corresponds to a $k$-fold cross validation where each of the $k$ folds represent a dataset to be tested, where the remaining folds are merged for training. b) It implies that potentially sensitive data needs to be shared across centres which poses a data security risk, and in practice requires a significant effort for properly anonymising the data.

Given our international, harmonised and annotated dataset, clinicians and researchers will be able to externally validate newly developed early warning systems for sepsis when considering the deployment in a new hospital. Compared to the internal validations, in our external validations we observed only a moderate decrease in PPV (or precision). However, alarm earliness did indeed suffer when applying a model to an unseen data distribution. Therefore, for deploying such an early warning system in a new site, we advise to fine-tune and recalibrate the model to the new target hospital, to already account for a changed (and plausibly unknown) prevalence of sepsis.

STRENGTHS AND LIMITATIONS Our study shows the following strengths. First, our multi-centre cohort includes a large sample size of ICU stays across three countries. Second, this cohort is composed of heterogeneous subcohorts. For instance, AUMC predominantly contains surgical patients, where the majority of patients in MIMIC-III and eICU represent medical patients. A third strength of this study was the depth of the conducted external validation, where we showcase that by using a federated learning approach, our models can generalise to unseen testing sites in new countries. Fourth and finally, we also consider the formulated prediction task a strength of our study. By simulating an online prediction scenario where a model is continuously updated with new input data in order to repeatedly output predictions, our scenario is more closely aligned and can be more realistically compared to a prospective deployment, in contrast to the retrospective horizon analyses that were conducted in the majority of the previous studies investigating the early prediction of sepsis [143]. Next, we discuss the limitations of this study. In this chapter, we presented an observational and retrospective study. This means that in order to derive clinical implications, we need to prospectively validate our findings by means of deploying the models in order to evaluate the clinical applicability and utility of bed-side predictions for sepsis. Next, due to poor data quality, a large number of patients (and even sites in the eICU dataset) were excluded, which could lead to selection effects. In two databases (eICU and HiRID), body fluid sampling was severely underreported, therefore an alternative definition of suspected infection (SI) was used. Even though we validated this definition, this modification may introduce a certain label shift between the centres which is hard to measure. Nevertheless, this could even increase the value of the external validation, since a good performing model may then be interpreted as also generalising despite minor label shifts. The harmonisation of variables across the datasets has the following limitations. In order to ensure interoperability, highly similar (but technically not identical) concepts were pooled[5]. Furthermore, to enable an interoperable setup allowing for the direct transfer of models across datasets, we determined a common ground in terms of time resolution as well as variable selection.

Furthermore, despite the broad inclusion criteria (essentially all non-pediatric ICU stays with recorded data) and the heterogeneity of the included cohorts, overall, our multi-centre dataset predominantly reflects a caucasian cohort. This deficit in ethnic diversity stems from the non-availability of non-caucasian ICU datasets, a pressing issue that needs to be urgently addressed by the global research community.

Finally, all the used patient data was collected before the outbreak of the global COVID-19 pandemic. Karakike et al. [102] found that the majority of hospitalised COVID-19 patients fulfil the clinical criteria of Sepsis-3. However, Sepsis-3 was neither developed nor validated (in terms of being prognostic of poor outcomes) for this cohort. This implies that in newly collected datasets (starting in 2019 or 2020, depending on the geographical region), COVID-19 patients fulfilling Sepsis-3 may need to be considered separately, and current sepsis prediction models need to be assessed in their ability to detect viral sepsis in COVID-19 patients.

## 4.5 CONCLUSION

In an international cohort of over 150,000 patient admissions to the ICU, we developed a deep learning-based early warning system for the early prediction of sepsis, that relies on routinely-collected data such as monitored vital signs and laboratory measurements. For the first time, in an extensive external validation across two continents, we demonstrated that a sepsis prediction model can indeed generalise to unseen hospitals internationally. It is our hope that the harmonised dataset as well as our conducted experiments will ultimately pave the way for the deployment and prospective validation of more robust and externally validated sepsis prediction systems.

---

[5] For instance, we did not differentiate between invasive and non-invasive blood pressure measurements.

# Part II

# Temporal and topological representation learning

# 5 PATH SIGNATURES FOR TIME SERIES REPRESENTATION LEARNING

We begin Part II of this thesis with a first chapter that considers representation learning on time series with *path signatures*. The content of this chapter is based on the following peer-reviewed workshop contribution:
M. Moor, M. Horn, C. Bock, K. Borgwardt, and B. Rieck. "Path Imputation Strategies for Signature Models". In: *ICML Workshop on the Art of Learning with Missing Values*. 2020

## 5.1 INTRODUCTION

Time series represent a class of data objects that are vested with a rich and complex structure, where subsequent measurements as well as different dimensions (or channels) can be correlated, and where streams of (incomplete) observations may arrive after irregular time intervals. Many machine learning (ML) models that try to leverage these data in order to arrive at predictions—be it a weather forecast, an alarm for a clinical complication, or a stock market price prediction—learn representations that summarise the irregular time series data in a vector that can be easily used for downstream tasks, e.g. to classify the time series. However, to properly encode a time series into vectors is non-trivial. To give an example, we may not be able to recover a time series from a set of summary statistics that were computed on its values (minimum, maximum, mean, variance, skewness, etc.).

In this chapter, we therefore consider *path signatures*, a framework that can be used to encode streams of temporal data at a negligible loss of information (see Section 5.2.1). However, the signature (or signature transform) is defined for continuous paths of data, as opposed to discrete time series samples. Specifically, the signature transform represents a universal non-linearity on the space of continuous paths that evolve in some Banach space such as $\mathbb{R}^n$. Provided such a continuous path, the signature returns a graded sequence of statistics that uniquely determines the path up to some negligible equivalence class [78]. It was first described by Chen in the 1950‘s [34, 35, 36], and then popularised in the theory of rough paths and controlled differential equations [59, 130, 131]. Over the last years, this transform has gained attention by the machine learning community for being a powerful feature extractor for analysing time series data [5, 124]. Recently, a model employing the signature won the 2019 PhysioNet Computing in Cardiology Challenge, showcasing the potential of applying the signature to clinical time series problems [145, 166].

When analysing real-world data processes, as opposed to continuous paths, we typically observe time series that are sampled merely at discrete points in time. Thus, to compute the signature, discrete time series measurements first need to be converted into a continuous path. Previous work treated this as an embedding problem, while considering it as a technical side note [19, 53]. In practice, this discrepancy is further disguised by the (sensible) way that software packages compute the signature: an input time series is interpreted as the knots of a piecewise linear path. Thus, when computing the signature, it is easy to mistakenly think of it as a function acting on discrete time series.

By contrast, in this work we show that the path construction can be relevant for achieving competitive performance when applying signature-based models to irregularly sampled time series. We consider the task of constructing a continuous path from discretely sampled input data as an imputation problem and refer to it as *path imputation*. While previous work has elaborated on various excellent theoretical properties of the signature [19, 38], we show that this does not necessarily translate to empirical performance. In our experiments, we investigate a variety of imputation strategies and compare several neural network architectures that may or may not employ the signature. Furthermore, motivated by coarseness of the default embedding (treating data as knots of a piecewise linear path), and given that data missingness itself can carry information[1], we also considered the flexible end-to-end learning framework of Gaussian process (GP) adapters, and propose an extension that enables uncertainty information to be exploited at individual prediction steps, which is beneficial for signature-based models.

## 5.2 The path signature

Before acquainting ourselves with the path signature, we first familiarise ourselves with some underlying concepts and notations.

**Definition 7** (Path)**.** *A* path $X$ *describes a continuous mapping from an interval to a real-valued vector space:*

$$\begin{aligned} X\colon [a,b] &\to \mathbb{R}^d \\ t &\mapsto X(t) \end{aligned} \tag{5.1}$$

---

[1] Already in this thesis, we have observed this in Chapter 3 and 4.

*for some* $t \in [a,b] \subseteq \mathbb{R}$*. For notational convenience, we allow for the decomposition of high-dimensional vector-valued paths into a collection of* $d$ *real-valued paths:* $X = \left(X^1, \ldots, X^d\right)$*, where* $X^i \colon [a,b] \to \mathbb{R}$*. Additionally, we abbreviate* $X(t) := X_t$*.*

**Definition 8** (Path integral)**.** *Let* $f \colon \mathbb{R} \to \mathbb{R}$ *be a function of a one-dimensional path* $X \colon [a,b] \to \mathbb{R}$*, then we define the* path integral *of* $X$ *against* $f$ *as*

$$\int_a^b f(X)\,\mathrm{d}X = \int_a^b f(X_t) \frac{\mathrm{d}X_t}{\mathrm{d}t}\,\mathrm{d}t, \tag{5.2}$$

*where the differential* $\mathrm{d}X = \frac{\mathrm{d}X_t}{\mathrm{d}t}\,\mathrm{d}t$ *can be intuitively understood as the rate of change of the path (w.r.t.* $t$*) that is applied for each infinitesimal increment* $\mathrm{d}t$*.*

Intuitively, the path integral measures (and accumulates) the changes in $f$ as we "walk" along a path $X$. Next, we consider a $d$-dimensional path $X$. We can then integrate along the dimension $i \in \{1, \ldots, d\}$ to obtain

$$S(X)^i_{a,t} := \int_{a<s<t} \mathbb{1}\,\mathrm{d}X^i_s = X^i_t - X^i_a, \tag{5.3}$$

where for clarity we indicate the integrand, the constant function $\mathbb{1}$, which is typically (and subsequently) omitted. Crucially, $S(X)^i_{a,\cdot} \colon [a,b] \to \mathbb{R}$ is itself also a real-valued *path*, $t \mapsto S(X)^i_{a,t}$, that is parametrised through $t \in [a,b]$. Therefore, we can iterate the integration along a dimension $j \in \{1, \ldots, d\}$ and with

$$S(X)^{i,j}_{a,t} := \int_{a<s<t} S(X)^i_{a,s}\,\mathrm{d}X^j_s = \int_{a<r<s<t} \mathrm{d}X^i_r\,\mathrm{d}X^j_s. \tag{5.4}$$

This can be further generalised to a collection of indices $i_1, \ldots, i_k \in \{1, \ldots, d\}$ with

$$S(X)^{i_1,\ldots,i_k}_{a,t} := \int_{a<s<t} S(X)^{i_1,\ldots,i_{k-1}}_{a,s}\,\mathrm{d}X^{i_k}_s, \tag{5.5}$$

$$:= \int_{a<t_k<t} \cdots \int_{a<t_1<t_2} \mathrm{d}X^{i_1}_{t_1} \ldots \mathrm{d}X^{i_k}_{t_k}, \tag{5.6}$$

where $S(X)^{i_1,\ldots,i_k}_{a,b}$ is called a *k-fold iterated integral* of $X$ along the indices $\{i_1, \ldots, i_k\}$ [38].

**Definition 9** (Path signature)**.** *Following Definition 7, let* $X = \left(X^1, \ldots, X^d\right) \colon [a,b] \to \mathbb{R}^d$ *be a path that is piecewise smooth. Then, the* path signature*, or simply the* signature*, is defined as the infinite collection of all its iterated integrals*

$$\mathrm{Sig}(X) := \left(1, S(X)^1_{a,b}, \ldots S(X)^d_{a,b}, S(X)^{1,1}_{a,b}, S(X)^{1,2}_{a,b}, \ldots S(X)^{d,d}_{a,b}, \ldots\right), \tag{5.7}$$

*where the superscripts increment over the set $I$ of all multi-indices*

$$I = \{(i_1, \dots i_k) \mid k \geq 1, i_1, \dots, i_k \in \{1, \dots, k\}\} \tag{5.8}$$

*in a specific order as indicated in Equation 5.7.*

Even though in Equation 5.7 we collected the terms of the form $S(X)_{a,b}^{i_1,\dots,i_k}$ in a flattened vector[2], the signature can actually be seen as an infinite collection of tensors, where each tensor comprises all components of Equation 5.7 that share the same number of indices in the superscript. The zeroth term belongs to $\mathbb{R}$, the first term to $\mathbb{R}^d \otimes \mathbb{R}^d$, the second term to $\mathbb{R}^d \otimes \mathbb{R}^d \otimes \mathbb{R}^d$ and so on, where $\otimes$ denotes the tensor product. This can be abbreviated as $\left(\mathbb{R}^d\right)^{\otimes k}$ where the product consists of $k$ terms, and by convention $\left(\mathbb{R}^d\right)^{\otimes 0} := \mathbb{R}$. Following this notation, the signature can be seen as an element of the tensor algebra of $\mathbb{R}^d$ which is defined as the direct product

$$T\left(\mathbb{R}^d\right) = \prod_{k \geq 0} \left(\mathbb{R}^d\right)^{\otimes k}. \tag{5.9}$$

In practice, we *truncate* the signature at depth $n$ by only computing the first $n$ tensors (without counting the constant zeroth term), that is, the first $\sum_{k=0}^{n} d^k$ entries of Equation 5.7.

**Definition 10** (Truncated signature)**.** *Following Definition 9, we define the truncated signature of depth $n$ (or up to the $n^{\text{th}}$ term) as*

$$\mathrm{Sig}^n(X) := \left( \int \dots \int_{a<t_1<\dots<t_k<b} \mathrm{d}X_{t_1} \otimes \dots \otimes \mathrm{d}X_{t_k} \right)_{0 \leq k \leq n}. \tag{5.10}$$

### 5.2.1 PROPERTIES OF THE SIGNATURE

The signature is rooted in a rich theoretical foundation, and is equipped with properties that make it an interesting transform for analysing time series. In the following, we state three of its key properties.

**Lemma 1** (Uniqueness of the signature [78])**.** *Let $X$ be a path (see Definition 7) that is piecewise smooth. Further, let $\hat{X}$ denote the time-augmented path of $X$ with*

$$\begin{aligned} \hat{X} \colon [a, b] &\to \mathbb{R}^{d+1}, \\ t &\mapsto (t, X_t). \end{aligned} \tag{5.11}$$

*The signature of the time-augmented path,* Sig$(\hat{X})$, uniquely *determines $X$ (up to translation).*

---

[2] This is also how software typically represents the truncated signature (see Definition 10).

Lemma 1 tells us that the signature allows us to embed a continuous path (in the tensor algebra) such that essentially no information is lost. Bonnier et al. [19] showed that in order to recover a target path $X$ using a randomly initialised template path $Y$, it is sufficient to minimise a norm between the signatures of $X$ and $Y$ by updating $Y$ via gradient descent. Nevertheless, certain aspects of an input path may not generally be recovered from its signature. For instance, the speed at which a path is traversed is not captured in the signature which is invariant under time reparametrisations. Additionally, the signature is blind with regard to certain degeneracies. For instance, it does not distinguish between a constant path and the concatenation of a path with its time-reversal [38]. Hambly et al. [78] established the uniqueness property demonstrating that the signature determines a path of bounded variation up to a "tree-like" equivalence class, a concept that makes the aforementioned degeneracies of self-intersecting paths precise.

**Lemma 2** (Factorial decay of higher-order terms [131])**.** *Let $X\colon [a,b] \to \mathbb{R}^d$ be a path, and let $\pi$ be a finite partition of $[a,b]$*

$$\pi = \{a = \pi_0, \pi_1, \dots, \pi_l = b\} \quad \textit{with} \quad \pi_i \leq \pi_j \quad \textit{for} \quad i < j. \tag{5.12}$$

*Furthermore, let $X$ be of bounded $p$-variation* $\mathrm{var}_p$*, i.e.,*

$$\mathrm{var}_p(X, \pi) = \sup\{\sum_{i=0}^{l-1} \|X(\pi_{i+1}) - X(\pi_i)\|^p\} < \infty. \tag{5.13}$$

*Then, the $k^{\text{th}}$ term of the signature decays* factorially *with regard to some tensor norm $\|\cdot\|_T$*

$$\left\| \int \cdots \int_{a<t_1<\cdots<t_k<b} \mathrm{d}X_{t_1} \otimes \cdots \otimes \mathrm{d}X_{t_k} \right\|_T \leq \frac{C(X)^{\frac{k}{p}}}{(\frac{k}{p})!}, \tag{5.14}$$

*where $C(X)$ is a constant that only depends on $X$.*

Lemma 2 implies that the higher-order terms of the signature contribute increasingly less to the signature. Finally, the signature exhibits the remarkable property of being a universal non-linearity on the space of paths.

**Lemma 3** (Universal non-linearity [4, 19])**.** *Let $X$ be a continuous path and $\hat{X}$ its time-augmented path (see Lemma 1). Let $F$ be any continuous function $F\colon \mathcal{X} \to \mathbb{R}$ defined on the space of continuous piecewise smooth paths $\mathcal{X}$ evolving in $\mathbb{R}^d$. Additionally, let $K$ denote a* *compact set of paths with* $X_0 = \mathbf{0}$, *if* $X \in K$. *For any* $\epsilon > 0$, *there exist a linear functional* $L$ *that fulfills*

$$\left| F(X) - L(\mathrm{Sig}(\hat{X})) \right| < \epsilon. \tag{5.15}$$

*Thus, the signature can approximate every function* $F$ *up to a linear transformation.*

### 5.2.2 COMPUTING THE SIGNATURE

The signature can be efficiently computed when using piecewise linear paths [109]. Algorithms to compute the signature exploit Chen's identity [131] which states that if $Z$ is the concatenation of two paths $X, Y\colon [a,b] \to \mathbb{R}^d$, with $X_b = Y_a$, then

$$\mathrm{Sig}(Z) = \mathrm{Sig}(X) \otimes \mathrm{Sig}(Y), \tag{5.16}$$

where $\otimes$ denotes the tensor product on the tensor algebra $T\big((\mathbb{R}^d)\big)$. This can be achieved by defining $\otimes$ such that for $A = (A_0, A_1, \dots)$ and $B = (B_0, B_1, \dots)$ with $A, B \in T\big((\mathbb{R}^d)\big)$

$$A \otimes B := \left( \sum_{j=0}^{k} A_j \otimes B_{k-j} \right)_{k \geq 0}. \tag{5.17}$$

If $L\colon [a,b] \to \mathbb{R}^d$ is the linear interpolation between two points $x, y \in \mathbb{R}^d$, then $\mathrm{Sig}(L)$ can be shown to equal the collection of powers of the increment $y - x$ [19, Section A.1], i.e.,

$$\mathrm{Sig}(L) = \left( 1, y - x, \frac{1}{2}(y-x)^{\otimes 2}, \dots, \frac{1}{6}(y-x)^{\otimes 3}, \dots, \frac{1}{k!}(y-x)^{\otimes k}, \dots \right). \tag{5.18}$$

By combining the Equations 5.18 and 5.16, we observe that the signature of a piecewise linear path $X$ with knots $(x_1, x_2, \dots x_n)$ can be directly computed using only tensor operations

$$\mathrm{Sig}(X) = \exp(x_2 - x_1) \otimes \exp(x_3 - x_2) \otimes \dots \otimes \exp(x_n - x_{n-1}), \tag{5.19}$$

where $\exp\colon \mathbb{R}^d \to T\big((\mathbb{R}^d)\big)$ denotes the exponential map on the tensor algebra

$$\exp(x) = \left( \frac{x^{\otimes k}}{k!} \right)_{k \geq 0}. \tag{5.20}$$

Equation 5.19 illustrates that the truncated signature can be computed by truncating the exponential map and then calculating the product $\otimes$ up to the available terms. Furthermore, it shows that the (truncated) signature may be implemented using basic tensor operations (without the need to invoke quadratures for high-dimensional integration), which has mo-

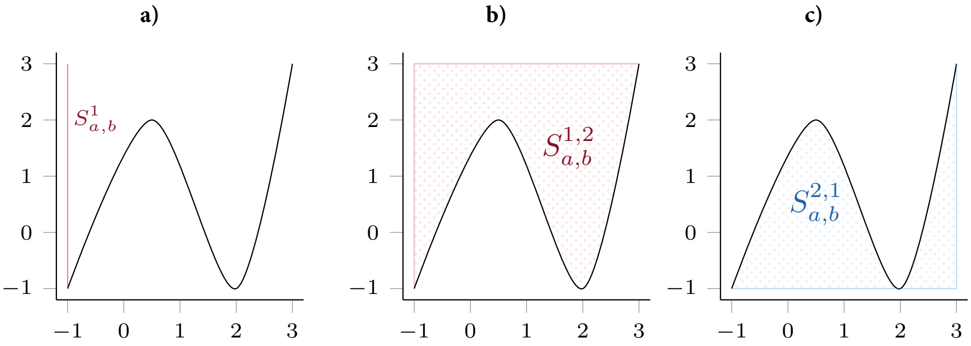


Figure 5.1: Visualisation of the signature for the first and second order terms. First-order terms are of the form $S^i_{a,b}$ and represent the *increment* of a path along dimension $i$. Second-order terms of the form $S^{i,j}_{a,b}$ and for $i \neq j$ correspond to signed areas above or below a path and are related to the Lévy area, i.e., the signed area between the path (evaluated from $a$ to $b$) and the chord (or linear interpolation) of $X_a$ and $X_b$.

tivated the development of libraries that allow for the computation of the signature as a differentiable layer in neural networks [109].

## 5.3 PATH IMPUTATION OF SIGNATURE MODELS

While the signature acts on continuous paths, we are typically only provided discrete measurements of data that may be irregularly spaced and, for a given point in time, incompletely observed. It is our working hypothesis, that the specific path construction from these data is relevant to the signature computation, and therefore to models employing the signature. To give a motivating example, Figure 5.1 depicts a geometric interpretation of the first and second order terms of the signature. Considering Figure 5.1b, it is evident that modifying this example path, for instance by only observing few discrete samples of it, and imputing missing values by carrying forward the last observed values, would affect the highlighted area—and therefore the signature—considerably. We assess this hypothesis by explicitly considering the path construction process as a *path imputation* problem.

TASK Let $\mathcal{T}$ be the space of time series (following Definitions 1 to 4) that are of regular spacing and fully observed, i.e., for each available point in time, each variable is observed. Furthermore, let $\mathcal{T}'$ denote the space of irregular (or sparse) time series, that allows for irregular spacing and potentially incomplete observations. Next, let $\mathcal{P}$ denote the space of continuous and piecewise linear paths. For a time series dataset (see Definition 2), which for some target space $\mathcal{Y}$ represents the finite subset $D \subset \mathcal{T}' \times \mathcal{Y}$, we consider the task to learn a composed mapping $g \colon \mathcal{T}' \to \mathcal{Y}$, with $g = f \circ \phi$, where $\phi \colon \mathcal{T}' \to \mathcal{P}$ represents the path imputation, and $f\colon \mathcal{P} \times \mathcal{W} \to \mathcal{Y}$ represents a classifier equipped with a parameter space $\mathcal{W}$[3]. Given a loss function $\ell$ and a set of $p$ path imputation strategies $\Phi = (\phi_1, \ldots, \phi_p)$, our task is to minimise

$$\underset{\phi_i \in \Phi, \mathbf{w} \in \mathcal{W}}{\arg\min} \quad \mathbb{E}_{(T', y) \sim \mathbb{P}(\mathcal{T}', \mathcal{Y})} \left[ \ell\big(g\big(T'; \phi_i, \mathbf{w}\big), y\big) \right]. \tag{5.21}$$

Thus, our goal is to learn path representations that are beneficial to a downstream classification tasks. To ensure a straight-forward computability of signatures, we treat a path imputation $\phi$ as the composition $\phi = \xi \circ \lambda$, where $\xi\colon \mathcal{T}' \to \mathcal{T}$ represent an imputation function mapping from the space of (potentially) irregularly spaced and incompletely observed time series, $\mathcal{T}'$, to the space of regularly spaced, fully observed time series, $\mathcal{T}$. $\lambda\colon \mathcal{T} \to \mathcal{P}$ maps a time series to a piecewise linear path, where the knots are defined by the input time series. For the scope of this work, $\lambda$ is fixed and shared for all path imputation strategies, which we introduce in the next paragraph.

Path imputation strategies In our analysis, we included a battery of path imputation strategies, that are subsequently described.

1. Linear interpolation: to impute a given point of time, we linearly interpolate between the previous and the next observation. If one of the two boundary points is missing, we impute them with 0 which after standardisation is equal to a mean imputation.

2. Forward filling: here, missing values are imputed with the last observed value of the given channel, whereas missing values at the start are imputed with 0.

3. Zero imputation: all missing values are imputed with 0, which equals a mean imputation when working with standardised data.

4. Indicator imputation: for each channel and point in time, we define a binary missingness indicator which is 1 if a given value is missing, and 0 else. The actual missing values are imputed with 0.

5. Causal imputation: here, we augment the time series by including additional observations such that we first update the time (of a new observation) while keeping the data of the previous point, and only then update the data (while keeping the time fixed)[4].

---

[3] This notation that classifier act on paths can be extended to classifiers that do not employ the signature, i.e., that do not by design act on paths, by considering the knots of the piecewise linear path as a sequence of discrete inputs.

[4] This method is related to the time-joined transformation [121]. For more details, please refer to [139, Section A.6].

6. Gaussian process (GP) adapters: we include both conventional GP adapters [125] (see Section 3.2.3), as well as an extension, GP adapter with posterior moments (GP-PoM), that we subsequently introduce.

While Strategies 1-5 are fixed transforms, the GP adapters represent an end-to-end learning framework, where both the imputation and downstream task are learned jointly.

Gaussian process adapters with posterior moments To revisit the GP adapter formulation, we refer to Chapter 3.2.3 and specifically to Equation 3.18. A drawback of existing GP adapters is the prediction step. The approximation of the expectation outside of the loss function in Equation 3.18 by means of Monte Carlo (MC) sampling is costly. To address this, Li et al. [125] proposed to sacrifice the uncertainty upon test time, by simply passing the GP posterior mean to the downstream classifier. However, since plugging a mean estimate of the GP is generally not equal to the GP adapter training objective (see Equation 3.18, our standard GP adapter follows Futoma et al. [61] by employing the more expensive (but uncertainty-preserving) MC sampling also upon testing time. However, even though the downstream classifier is called $n_m$ times (for $n_m$ MC samples), at each prediction step, the model sees only a single draw and is therefore unaware of any uncertainty in the GP imputation. The uncertainty-awareness in the GP adapter that uses MC sampling only emerges on a meta level: by making predictions for several independently drawn samples, the distribution of prediction scores can be interpreted as a measure of uncertainty about the classifiers prediction.

Here, we address both points with a novel variant of a GP adapter where both moments of the posterior distribution (mean and covariance) are provided to the classifier. On the one hand, this prevents the cost of MC sampling, while on the other hand still providing uncertainty information, and as opposed to the standard GP adapter, making uncertainty directly available to the classifier when running a single prediction (without repeated sampling). While the full covariance matrix may quickly become excessively large without the guarantee that all interaction terms are actually relevant to a downstream classifier, we for now simplify this approach by taking only the posterior *variance* at each location of the GP, to concatenate it with the posterior mean in order to produce a path which also comprises point-wise uncertainty information. Denoting the hyperparameters of the GP with $\boldsymbol{\theta} \in \Theta$ for a hyperparameter space $\Theta$, we define a mapping $\tau$ with

$$\tau \colon [a, b] \times \mathcal{T}' \times \Theta \to \mathcal{T} \times \mathcal{T} \tag{5.22}$$

$$\tau \colon t, T', \boldsymbol{\theta} \mapsto \big(\mu(t, T'; \boldsymbol{\theta}), \Sigma(t, t, T'; \boldsymbol{\theta})\big). \tag{5.23}$$

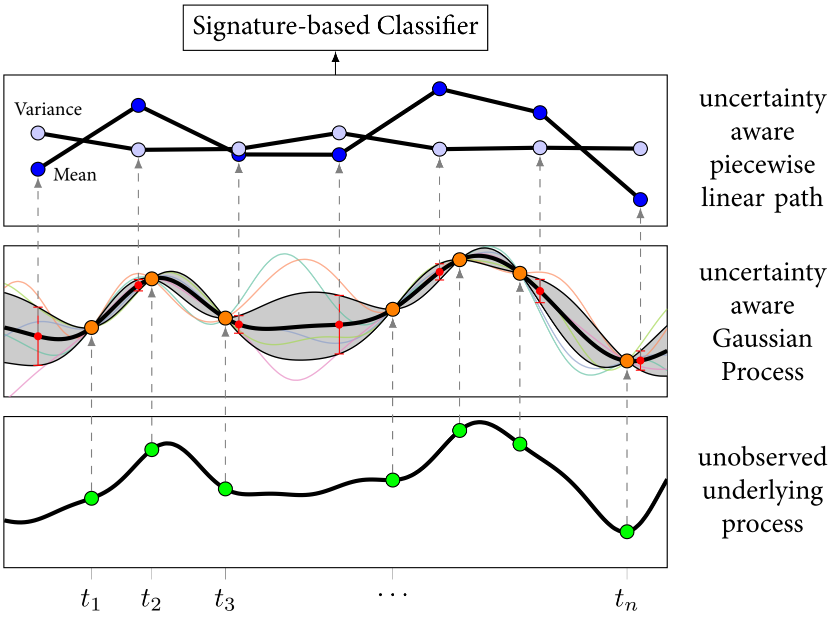

Signature-based Classifier
Variance
Mean
uncertainty aware piecewise linear path
uncertainty aware Gaussian Process
unobserved underlying process
$t_1$
$t_2$
$t_3$
$\cdots$
$t_n$

Figure 5.2: Overview of the GP adapter with posterior moments (GP-PoM), our extension to GP adapters that upon testing time leverages both posterior moments (mean and variance) of a given test location. In comparison, to achieve uncertainty in the conventional GP adapter, MC samples (faded colours in the background) are drawn from the GP posterior and fed into the downstream classifier.

Using $\tau$, we then solve

$$\underset{\mathbf{w}\in\mathcal{W},\boldsymbol{\theta}\in\Theta}{\arg\min} \sum_{k=1}^{N} \ell(F(\tau(\,\cdot\,, T'_k, \boldsymbol{\theta}), \mathbf{w}), y_k). \tag{5.24}$$

As stated above, we decompose path imputations such that $\phi = \xi \circ \lambda$. Therefore, in this setting, to construct a time series imputation $\xi$, we evaluate $\tau$ only at discrete, regularly spaced time steps. Finally, we call this method GP adapter with posterior moments (GP-PoM) and illustrate it in Figure 5.2.

## 5.4 RELATED WORK

A key motivation for considering the signature as a way to learn time series representations is its increasing use in machine learning [19, 38, 113, 115, 124, 145] that is founded in its favourable theoretical properties (see Section 5.2). While the signature was conventionally employed as a non-parametric feature extractor, recent works have been investigating how to integrate the signature into neural networks [109, 126]. Furthermore, Király et al. [113]

showed how the truncated signature can be used to define a *kernel*, which led Toth et al. [199] to define a Gaussian process using signature covariances. Most recently, Salvi et al. [180] showed how even an untruncated signature kernel can be computed. The literature has typically applied the linear interpolation in order to encode discrete data samples as paths. While some slight deviations from this default have been considered [121], to our knowledge no previous work has regarded and empirically investigated the impact of different choices of path imputations.

IMPUTATION SCHEMES The imputation of missing data is a well-studied statistical problem, see for example [66, Chapter 25]. However, imputation methods typically only fill in a discrete set of missing values of data, and do not consider the problem of imputing a continuous process that underlies the data. In contrast, Gaussian process adapters [125] are capable of imputing a continuous path that can be arbitrarily sampled, which makes them particularly amenable to be used in our analysis. There are also other approaches that learn to impute missing data end-to-end with a downstream classifier [184], and methods that skip explicit imputations altogether, be it with Neural ODE-like models [110, 175], recurrent neural network architectures [32], or SeFT, a set functions approach to time series [93] (the last of which was co-authored by the author of this thesis). Since the scope of this chapter was to assess the impact of path imputations for the signature (and models employing the signature), the larger comparison including imputation-free scenarios will be interesting in future work, while not essential to the central idea of this chapter.

LEARNING TIME SERIES REPRESENTATIONS The signature transform exhibits a certain similarity with other well-established transforms that are ubiquitous in signal processing and various application domains, such as the Wavelet transform [136], or the Fourier transform [22]. In all of them, we compute integrals over paths to retrieve a representation of data in a new domain (frequency domain, tensor algebra etc.). However, at a closer look, there are striking differences between the signature and these classical transforms: The Fourier and Wavelet transforms are *linear* transforms, and act on each channel of the input data separately, i.e., they model an input path as a linear combination of the elements of a basis. By contrast, the signature is a non-linear transform (even a universal non-linearity), and actually combines information between different channels. By doing so, instead of providing a basis for paths, the signature can be seen as providing a basis for *functions of paths*.

In addition to these aforementioned transforms that may be seen as fixed feature extractors, recent advances in deep learning have brought forth a bouquet of model classes for *learning* to transform time series data into representations that best serve a given downstream purpose. To name a few, this includes studies for learning time series representations using convolutional autoencoders [90], temporal convolutional networks [195], GP-VAE [58], latent ODEs [175], and most recently, score-based diffusion models [193]. Since the signature can be directly integrated as a layer of a neural network, in principle, it could augment any of the above approaches.

## 5.5 EXPERIMENTS

DATASETS AND PREPROCESSING To evaluate different path imputation strategies, we consider the classification of time series as our downstream task, which we present for two real-world datasets: (i) PENDIGITS [50], and (ii) CHARACTERTRAJECTORIES [50]. For the PENDIGITS dataset, we count 10,992 samples, featuring 2 channels and 8 time steps, and 10 classes. The CHARACTERTRAJECTORIES dataset contains 2,858 instances, featuring 3 channel dimensions, 182 time steps and 20 classes. We investigated two types of irregular sampling, by subsampling the time series in the following manner:

a) 'Random' subsampling (Missing at random): on the instance level, we randomly discard $p$% of all observations.

b) 'Label-based' subsampling (Missing not at random): for each class, we uniformly sample a class-specific missingness frequency between $p - 10\%$ and $p + 10\%$.

For CHARACTERTRAJECTORIES, we use $p = 50$, and for PENDIGITS which consists of short time series, we use more moderate subsampling frequencies with $p = 30$. All time series were $z$-scored using the empirical moment estimates that were determined on the entire training split (for more details regarding the splits, please refer to the paragraph Training and evaluation below).

MODELS In our experiments, we investigate the following model architectures:

a) SIG, a simple neural network that involves a linear augmentation, followed by the signature transform (signature block) and a final MLP of two dense layers $(30, 30)$. This architecture was inspired by the Neural-signature-augment model [19].

b) RNN, a recurrent neural network employing gated recurrent units (gru) [40],

c) RNNSIG, where the signature transform is computed in a sliding window fashion resulting in a stream of signatures, which is then processed by a gru, and

d) DeepSig, a deep signature model that sequentially applies two signature blocks (each comprising an augmentation step and one signature transform). This architecture was inspired by the DeepSigNet [19].

To efficiently compute the signature transform on the GPU, we used the 'Signatory' package [109]. All GP adapters were implemented based on (and compatible with) the 'GPyTorch' framework [63].

Training and evaluation For each dataset, we used the predefined training and testing sets where 20% of the training set was held out as a validation set for tuning the hyperparameters. For each element of the grid (dataset × model × path imputation), we ran a randomised search over 20 configurations of hyperparameters. Each run was trained until convergence, i.e., training was stopped if the validation performance did not improve over 20 epochs, or if 100 epochs were reached. These datasets represent a multi-class classification task which we optimised in terms of balanced accuracy (BAC). Additionally, we report accuracy and weighted AUROC (w-AUROC), where AUROC is calculated for each class (using a one-versus-one strategy) and averaged with weights that correspond to the support of each class. Having tuned the hyperparameters for each setting on the aforementioned grid of experiments, we retrained 5 model repetitions and selected the best model state in terms of the validation BAC to finally report mean and standard deviation of the evaluation metrics on the testing set.

Results We present our results in Tables 5.1 and 5.2 under label-based subsampling. For the random subsampling, please refer to Tables A.1 and A.2 in the appendix. We find that DeepSig as well as the RNN perform well in many scenarios, suggesting they are impervious to the choice of path imputation strategy. While this robustness may be somewhat unsurprising for the RNN which does rely on paths, the robustness of the deep signature model is less obvious. In contrast, when comparing the last rows across all table blocks, we observe that the shallow signature model, Sig, is heavily impacted by the choice of path imputation scheme.

We visualise this finding in Figure 5.3, where we depict the results for the CharacterTrajectories dataset. Notably, exactly for shallow signature models, that we find to be more sensitive to changes in the path construction, we also observe that our GP-PoM strategy leads to more robust behaviour of the models in terms of a beneficial performance. In PenDigits, for the standard GP adapter (but not so for GP-PoM) we encountered numerical stability issues (which were addressed by jittering the diagonal in

Table 5.1: `CharacterTrajectories` dataset under label-based subsampling. The top three methods are highlighted: bold & underlined, bold, underlined. All measures are reported as percentage points. Balanced accuracy (BAC) is the metric we optimised for. We further report accuracy and weighted AUROC (w-AUROC).

| Imputation | Model | w-AUROC | BAC | Accuracy |
|---|---|---|---|---|
| GP-PoM | DeepSig | 99.582 ± 0.671 | 95.155 ± 1.501 | 94.958 ± 1.716 |
| | RNN | 99.973 ± 0.015 | **98.161 ± 0.664** | **98.273 ± 0.602** |
| | RNNSig | 99.696 ± 0.089 | 92.778 ± 1.239 | 93.231 ± 1.133 |
| | Sig | 99.516 ± 0.075 | 88.627 ± 1.416 | 89.011 ± 1.319 |
| GP | DeepSig | 99.290 ± 0.704 | 89.545 ± 2.996 | 89.368 ± 3.123 |
| | RNN | 99.970 ± 0.011 | 97.712 ± 0.266 | 97.873 ± 0.251 |
| | RNNSig | 96.669 ± 2.393 | 65.717 ± 13.691 | 67.052 ± 13.182 |
| | Sig | 95.283 ± 1.602 | 62.423 ± 6.110 | 63.614 ± 5.958 |
| causal | DeepSig | 99.940 ± 0.024 | 97.272 ± 0.709 | 97.437 ± 0.620 |
| | RNN | 99.960 ± 0.010 | 97.239 ± 0.516 | 97.409 ± 0.481 |
| | RNNSig | 99.523 ± 0.155 | 89.922 ± 2.301 | 90.585 ± 2.186 |
| | Sig | 95.747 ± 4.957 | 66.307 ± 21.794 | 68.259 ± 20.757 |
| forward-filling | DeepSig | 99.953 ± 0.041 | 97.956 ± 0.677 | 98.078 ± 0.656 |
| | RNN | 99.942 ± 0.011 | 96.942 ± 0.486 | 97.159 ± 0.444 |
| | RNNSig | 99.720 ± 0.071 | 92.568 ± 1.091 | 93.148 ± 1.011 |
| | Sig | 94.828 ± 8.117 | 67.169 ± 26.338 | 68.649 ± 26.125 |
| indicator | DeepSig | **99.988 ± 0.013** | **98.591 ± 0.294** | **98.719 ± 0.263** |
| | RNN | 99.916 ± 0.020 | 96.414 ± 0.406 | 96.671 ± 0.367 |
| | RNNSig | 99.802 ± 0.032 | 93.787 ± 0.463 | 94.234 ± 0.442 |
| | Sig | 91.661 ± 10.003 | 56.423 ± 22.796 | 58.384 ± 22.932 |
| linear | DeepSig | 99.970 ± 0.010 | 98.051 ± 0.743 | 98.217 ± 0.671 |
| | RNN | 99.880 ± 0.059 | 96.906 ± 1.314 | 97.117 ± 1.196 |
| | RNNSig | 99.876 ± 0.035 | 94.848 ± 0.916 | 95.292 ± 0.842 |
| | Sig | 80.442 ± 18.228 | 31.193 ± 23.962 | 32.326 ± 24.679 |
| zero | DeepSig | **99.977 ± 0.010** | 98.030 ± 0.357 | 98.189 ± 0.358 |
| | RNN | 99.967 ± 0.014 | 97.428 ± 0.572 | 97.549 ± 0.596 |
| | RNNSig | 99.699 ± 0.132 | 91.752 ± 1.782 | 92.368 ± 1.662 |
| | Sig | 77.727 ± 23.671 | 37.992 ± 34.456 | 38.955 ± 35.232 |

Table 5.2: `PenDigits` dataset under label-based subsampling. The top three methods are highlighted: bold & underlined, bold, underlined. All measures are reported as percentage points. Balanced accuracy (BAC) is the metric we optimised for. We further report accuracy and weighted AUROC (w-AUROC)

| Imputation | Model | w-AUROC | BAC | Accuracy |
|---|---|---|---|---|
| GP-PoM | DeepSig | 99.930 ± 0.032 | **97.403 ± 0.300** | **97.381 ± 0.298** |
| | RNN | 99.901 ± 0.016 | 96.349 ± 0.297 | 96.306 ± 0.302 |
| | RNNSig | 99.669 ± 0.073 | 93.022 ± 0.765 | 92.967 ± 0.763 |
| | Sig | 99.150 ± 0.144 | 88.090 ± 1.493 | 87.999 ± 1.499 |
| GP | DeepSig | 92.885 ± 1.455 | 60.593 ± 4.092 | 60.476 ± 4.067 |
| | RNN | 95.170 ± 1.438 | 67.543 ± 4.782 | 67.426 ± 4.790 |
| | RNNSig | 84.501 ± 1.307 | 42.184 ± 1.977 | 42.141 ± 1.913 |
| | Sig | 80.312 ± 2.655 | 37.767 ± 3.611 | 37.725 ± 3.646 |
| causal | DeepSig | 99.241 ± 0.075 | 89.616 ± 0.749 | 89.514 ± 0.747 |
| | RNN | 99.241 ± 0.098 | 89.496 ± 0.480 | 89.417 ± 0.501 |
| | RNNSig | 99.298 ± 0.041 | 89.187 ± 0.476 | 89.137 ± 0.494 |
| | Sig | 98.374 ± 0.065 | 83.205 ± 0.404 | 83.082 ± 0.426 |
| forward-filling | DeepSig | 99.007 ± 0.072 | 88.205 ± 0.434 | 88.090 ± 0.428 |
| | RNN | 99.333 ± 0.046 | 89.747 ± 0.406 | 89.657 ± 0.419 |
| | RNNSig | 99.274 ± 0.015 | 89.788 ± 0.384 | 89.743 ± 0.392 |
| | Sig | 98.310 ± 0.045 | 83.739 ± 0.421 | 83.625 ± 0.398 |
| indicator | DeepSig | **99.960 ± 0.013** | **98.068 ± 0.184** | **98.056 ± 0.185** |
| | RNN | **99.955 ± 0.009** | 97.266 ± 0.439 | 97.238 ± 0.447 |
| | RNNSig | 99.747 ± 0.028 | 93.488 ± 0.616 | 93.408 ± 0.613 |
| | Sig | 99.410 ± 0.031 | 90.591 ± 0.306 | 90.492 ± 0.308 |
| linear | DeepSig | 99.458 ± 0.052 | 91.567 ± 0.412 | 91.452 ± 0.416 |
| | RNN | 99.489 ± 0.093 | 91.608 ± 0.609 | 91.492 ± 0.608 |
| | RNNSig | 99.446 ± 0.039 | 90.259 ± 0.859 | 90.143 ± 0.869 |
| | Sig | 98.963 ± 0.084 | 87.254 ± 0.437 | 87.141 ± 0.458 |
| zero | DeepSig | 99.391 ± 0.071 | 91.121 ± 0.406 | 91.012 ± 0.403 |
| | RNN | 99.551 ± 0.031 | 91.765 ± 0.283 | 91.670 ± 0.304 |
| | RNNSig | 99.321 ± 0.033 | 89.543 ± 0.412 | 89.457 ± 0.417 |
| | Sig | 98.544 ± 0.069 | 84.269 ± 0.445 | 84.185 ± 0.454 |

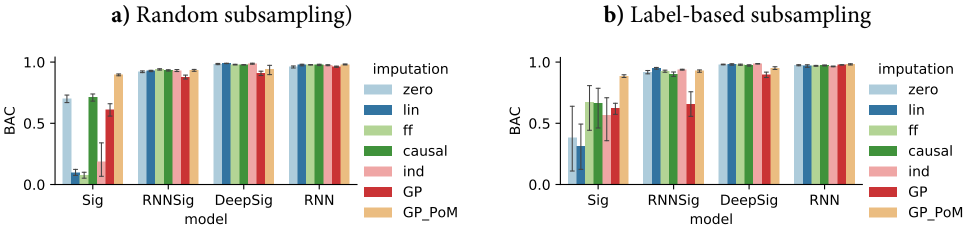


Figure 5.3: Visual depiction of the results for the CharacterTrajectories dataset. The bars indicate the performance in terms of balanced accuracy (BAC). The panels indicate the subsampling strategy. Left: random subsampling, right: label-based subsampling.

the Cholesky decomposition). In general, we observed that GP-PoM tends to converge more quickly to a better performance than the standard GP adapter (see Figure A.19).

## 5.6 DISCUSSION

In this chapter, we have considered the path signature, a theoretically well-founded transform, that may be used as a neural network layer to learn representations of time series. With regard to applying signatures—that act on continuous paths—to real-world discretely sampled time series data, we stated the hypothesis that the exact choice of path construction could impact the resulting signature and models that use it downstream. To this end, and given by the constraints to efficiently compute the signature (via piecewise linear paths), we formulated this task as a path imputation problem, and included a battery of imputation strategies including a novel extension of a Gaussian process adapter, GP-PoM.

In our experiments, we found that the choice of path imputation scheme can indeed drastically affect the performance of signature-based models. Most prominently, we observed this effect for shallow signature models, while deeper signature models were more resilient and robust in tackling irregularly spaced and incompletely observed multivariate time series over different path imputations.

Furthermore, we found that approaches that are aware of data missingness (GP-PoM and indicator imputation) were beneficial for constructing paths from raw time series data. Our experiments confirm that uncertainty information has to be accessible during the *prediction step*. We highlighted that this is indeed not the case for the original GP adapter (despite the name "uncertainty-aware framework"), since for each MC sample, the classifier cannot access missingness information or uncertainty about the underlying imputation.

GP-PoM, our proposed end-to-end imputation strategy, lead to competitive performance, and improved upon the conventional GP adapter. As a limitation, GP-PoM sacrifices the innate ability of the GP adapter to remain uncertain *about* its own prediction (via the variation of predictions over different MC samples).

Recommendations The signature is a powerful transform that is able to encode paths at almost no loss of information [38]. The recent literature brought forth examples, where signature-based models lead to stellar empirical performance [19, 145]. Therefore, we recommend to consider this transform, when classifying time series. In this work, we demonstrated that applying the signature to (irregularly sampled) time series comes a certain cost: the necessary construction of a continuous path from discrete time series observations can be a delicate task that can adversely impact the signature as well as downstream models. To address this, we recommend GP-PoM, which explicitly captures a continuous degree of uncertainty in the imputed path which is made available to the downstream model at each prediction step. Furthermore, the indicator imputation exhibited convincing performance and presented as a simple and promising go-to solution. However, we caution its use in shallow signature models where we observed a detrimental impact on the performance.

Applying signature models in online prediction tasks, where to mimic the testing scenario already during training data should not leak from the future, we recommend to consider causal (or time-joined) path imputations. The main idea behind this approach is to prevent future data leakage despite using piecewise linear paths, i.e., paths that are linearly interpolated between observations.

## 5.7 Concluding remarks

The path signature has gained attention in the machine learning community for being a powerful feature extractor that can be seamlessly integrated into neural networks as a differentiable layer. Here, we hypothesised and also empirically demonstrated that the application of the signature to real-world time series is fraught with pitfalls—we found the choice of the path imputation strategy to be essential for obtaining high predictive performance, in particular in shallow models, whereas deeper signature models were more robust.

Furthermore, with GP-PoM, we made uncertainty information available to the prediction step which has led to competitive performance in general, and improved the robustness of shallow signature models, in particular.

# 6 TOPOLOGICAL REPRESENTATION LEARNING

In this second chapter of Part II, the part of the dissertation that is concerned with representation learning, we develop a novel deep learning framework that enables us to learn data representations that preserve the structure of the original, potentially high-dimensional input data. The content of this chapter are based on the following publication:

M. Moor$^{\dagger}$, M. Horn$^{\dagger}$, B. Rieck$^{\ddagger}$, and K. Borgwardt$^{\ddagger}$. "Topological Autoencoders". In: *International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054

In the preceding Chapter 5, we have shifted our focus away from clinical applications of time series classification (see Part I), in favour of attending more closely to the inner workings and learned representations of deep neural networks that classify time series. In this chapter, we go a step further and explore how we can learn meaningful data representations in a more general context. Depending on the downstream use, learned representations need to exhibit different properties. For instance, to maximise classification performance, representations that maximise class separability are most useful. In contrast, when performing dimensionality reduction, data visualisation, or data exploration, it can be more desirable to learn representations that faithfully reflect the structure of potentially high-dimensional input data. In this chapter, we follow the second route, and propose a novel method for learning embeddings that preserve the multi-scale structural features of the input space as measured with techniques from topological data analysis.

This chapter is organised as follows: Section 6.1 gives a high-level introduction to the chapter. We continue by introducing some preliminaries in topological data analysis in Section 6.2 and 6.3. Building on this, we then present our novel method, topological autoencoder (TopoAE), and elaborate on its theoretical properties in Section 6.4. Subsequently, we put this method into context with the existing literature in Section 6.5, and conduct empirical experiments in Section 6.6, followed by a discussion of our findings in Section 6.7.

## 6.1 INTRODUCTION

Multi-scale topological features, as computed via persistent homology (see Section 6.3), have been increasingly used by the machine learning community [26, 77, 87, 88, 89, 161, 165]. However, while taking the topological perspective may add a rich set of structural information to the machine learning pipeline, it has been notoriously hard to directly optimise a model that includes topological computations. These circumstances are rooted in the fact that topological computations are typically of discrete nature, which makes it challenging to incorporate them in an end-to-end differentiable model. While conventionally, topological signatures were treated as fixed features, initial efforts to differentiate through topological calculations were made possible under specialised circumstances [33, 88, 159].

Here, we present a novel approach that allows us to compute gradients of topological signatures that allows us to employ topological constraints during the training of deep neural networks in order to learn low-dimensional representations that preserve the complex structure of the input data.

This chapter makes the following contributions:

- We propose topological autoencoder (TopoAE) that includes a novel topological loss term that allows to harmonise the structure (in terms of topological features) of a learnt latent space with the structure of the data space.
- By proving that the proposed loss term is robust on the level of mini-batches, our approach can be easily scaled to large datasets, where an explicit calculation of persistent homology of the entire dataset becomes infeasible.
- We show that our novel loss term leads to favourable embeddings by learning to preserve topological structures of complex input data in low-dimensional representations.

## 6.2 WHAT IS TOPOLOGY?

Topology describes the mathematical discipline that studies connectivity properties in a class of spaces referred to as *topological spaces*. A topological space is a set of points that live in a space such as $\mathbb{R}^n$ while being equipped with a notion of distance (or connectedness) between points. But before diving into more preliminaries, we first build some intuition what topology is about with a historic example. As probably one of the earliest contributions to this field, Leonhard Euler's "Seven brigdes of Königsberg" describe a famous problem where the question is whether it is possible to traverse the city of Königsberg by crossing each bridge exactly once such that one ultimately arrives at the starting point again. Euler exploited that most of the geometric information available on a map is actually irrelevant for solving this puzzle. Therefore, he distilled the problem into an abstracted graph, where each vertex represents a disconnected land mass of the city and each edge refers to a bridge, thereby considering only the connectivity information of the problem. Euler then found that for a desired path to exist, for a given land mass there needs to be one way in and one way out of it, thus requiring an even degree for each vertex of the graph. However, since this was not the case (the vertices had only odd degrees), Euler showed that the problem cannot have a solution [169].

Meanwhile, contemporary topology has grown into a rich field, but still, connectivity information is of central interest. Having laid the ground for what topology is about, we next consider some preliminaries.

While the concepts we make use of in this chapter can operate over highly generic spaces, for our purposes it is sufficient to think of $\mathbb{R}^n$ as the 'prototypical' topological space[1]. Topology focuses on connectivity information. Therefore, objects (i.e., topological spaces) that may geometrically look different, can from a topological perspective be considered equal, or more precisely, *homeomorphic*.

**Definition 11** (Homeomorphism). *Let $X, Y \subseteq \mathbb{R}^n$ be two topological spaces. Then the mapping $f\colon X \to Y$ is a* homeomorphism *if $f$ is bijective, continuous, and its inverse $f^{-1}$ is also continuous.*

As a famous illustrative example for a homeomorphism, we may consider a solid torus and a mug[2]. If the materials were flexible enough, we could transform the objects into each other without tearing or cutting them at any place, making the two objects indistinguishable from a topological perspective. To characterise properties of topological spaces that are invariant under smooth, homeomorphic transformations (as illustrated in the above example), one can consider Betti numbers, a concept from simplicial homology. Informally, Betti numbers count the number of $d$-dimensional holes a space has[3], where $d = 0$ refers to the number of connected components, $d = 1$ to the number of cycles, $d = 2$ to the number of voids, and so on. To give two examples considering the first three dimensions: the 2-sphere $S^2 = \{x \in \mathbb{R}^3 \mid \|x\| = r\}$ (for any radius $r > 0$) has the Betti numbers $(1, 0, 1)$ as it comprises one connected component that encloses a single void. The (hollow) 2-torus $T^2 = S^1 \times S^1$, which can be defined as the Cartesian product of two circles, shows the Betti numbers $(1, 2, 1)$, i.e., it contains one connected component, two cycles, and one enclosed void. However, considering real-world datasets is fundamentally different from the discussed

---

[1] This way we do not need to introduce concepts like continuity in an abstract sense (via open sets), but may use our usual understanding of continuity from analysis.

[2] Alternatively, we may use a torus and a hollow mug

[3] Formally, the $d^{\text{th}}$ Betti number represents the rank of the $d^{\text{th}}$ homology group.

examples for two main reasons: 1. The data manifold is generally unknown, which is why invariants like the Betti numbers can not be directly determined from the manifold structure. 2. Real-world data typically consists only of discrete data samples that may be arranged close to a smooth manifold.

The first problem can be addressed with simplicial homology, a method that allows for the calculation of Betti numbers by applying matrix reduction algorithms to a particular representation of the data, the so-called simplicial complex, which we subsequently introduce.

**Definition 12** (Abstract simplex)**.** *Given a set of sets, any subset of cardinatlity $k$ is called a $k$-simplex. In contrast to a geometric $k$-simplex, which is a polytope fullfilling further properties (affinely independent points), the abstract simplex is defined more generally for sets, i.e., any constellation of vertices.*

**Definition 13** (Abstract simplicial complex)**.** *An abstract simplicial complex $\mathfrak{K}$ is a finite set of simplices that fulfils the following properties:*

*i) If $A$ is a simplex of $\mathfrak{K}$, then every face (i.e., subset) of $A$ is also in $\mathfrak{K}$.*

*ii) The non-empty intersection $A \cap B$ of any two simplices $A, B \in \mathfrak{K}$ is a face of both $A$ and $B$.*

By representing data points as a simplicial complex where connectivity between points is typically determined via some notion of distance or similarity, simplicial homology allows for the computation of Betti numbers, also when the data manifold is not known. Formally, the $d^{\text{th}}$ Betti number represents the rank of the $d^{\text{th}}$ homology group $H_d(\mathfrak{K})$ of the simplicial complex $\mathfrak{K}$ with respect to a boundary homomorphism. For more details, we refer the technically interested reader to Moor et al. [141, Section A.1]. However, even if we could compute Betti numbers from real-world data, we still have to address the second problem, i.e., the fact that we are typically given only discrete data samples and not observing smooth manifolds directly. In Figure 6.1, with the example of the torus we exemplify how this can be problematic.

If instead of the torus (Fig 6.1a), we are provided with samples (Fig 6.1b), it is not a priori clear which is the proper *scale* of the data, i.e., at which scale two neighbouring points should be considered connected. For instance, were we to connect all points that are closer than a threshold $\epsilon$, then the specific value of $\epsilon$ determines whether or not we can capture the interesting Betti numbers of the torus. If $\epsilon$ is chosen too small, the cycles and voids may not be recovered due to too sparsely distributed simplices. In contrast, if $\epsilon$ is too large, the resulting simplicial complex would become homeomorphic to a point, where again the voids and cycles were lost.

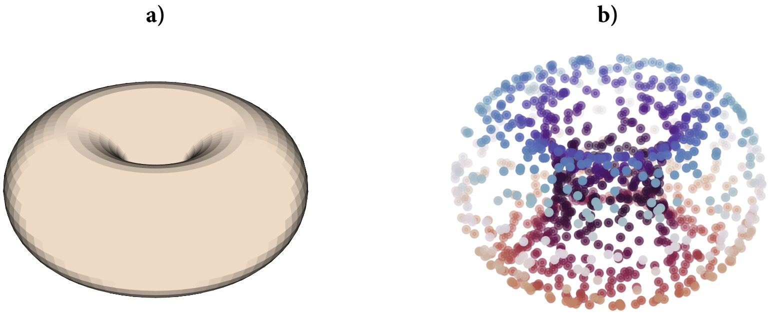


Figure 6.1

This illustrates that simplicial homology is too brittle and unstable for applying it directly to real-world data, since it is challenging to know the "proper" simplicial complex in advance. This has motivated *persistent homology* [9, 51], a method that allows for the calculation of topological features over multiple scales.

## 6.3 PERSISTENT HOMOLOGY: TOPOLOGY AT MULTIPLE SCALES

In persistent homology, we aim to calculate topological features (such as Betti numbers) at various scales. First, to intuitively motivate why multiple scales can be helpful, let us consider a point cloud $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^3$ of points that are uniformly sampled on the 2-sphere with some i.i.d. additive noise. Also, we equip ourselves with a metric such as the Euclidean distance. As we have noticed in the previous section, for retrieving the characteristic features (here the void), we need to consider the proper scale. For this, we use the metric to define $\epsilon$-balls $B_\epsilon(x_i) = \{p \in \mathbb{R}^3 | \, \|p - x_i\| \leq \epsilon\}$ around each point $x_i \in X$. For a given $\epsilon > 0$, we consider all simplices (subsets of points) for which $\epsilon$ upper bounds the diameter of the simplex (see Definition 14). By collecting all these simplices, we construct a scale-specific simplicial complex, the Vietoris–Rips complex (VR) complex [203] for scale $\epsilon$ which we denote as $\mathfrak{R}_\epsilon(X)$ (see Definition 14). Now, if we grow $\epsilon$ from $0$ to the value of the largest pairwise distance in $X$ and construct a $\mathfrak{R}_\epsilon(X)$ at each $\epsilon$ and calculate its Betti numbers, we find that the large void of the 2-sphere appears as soon as $\epsilon$ is large enough that the void is fully enclosed by the included simplices. Also, once the $\epsilon$ scale is large enough that the void vanishes, the corresponding Betti number becomes $0$ again. It is generally assumed that relevant topological features *persist* over a large range of scales, whereas features that appear and disappear shortly thereafter, are considered as noise.

**Definition 14** (Vietoris–Rips complex)**.** *For a finite metric space* $(\mathcal{S}, \mathrm{d})$*, the Vietoris–Rips complex at a scale* $\epsilon$ *is defined as the simplicial complex that contains simplices* $\sigma$ *that fulfil:*

$$\begin{aligned}\mathfrak{R}_\epsilon(\mathcal{S}) &:= \{\sigma \subseteq \mathcal{S} \mid \operatorname{diam}(\sigma) \leq \epsilon\}, \textit{ where} \qquad (6.1)\\ \operatorname{diam}(\sigma) &= \sup\{\mathrm{d}(s_i, s_j) \mid s_i, s_j \in \sigma\}. \qquad (6.2)\end{aligned}$$

In the above example, over varying scales $\epsilon$ we created simplicial complexes. This process of constructing a nested sequence of simplicial complexes of the form

$$\varnothing = \mathfrak{K}_0 \subseteq \mathfrak{K}_1 \subseteq \cdots \subseteq \mathfrak{K}_{m-1} \subseteq \mathfrak{K}_m = \mathfrak{K} \qquad (6.3)$$

is called a *filtration*. To further complement the above description of such a filtration, Figure 6.2 gives a visual illustration of a filtration of a point cloud. Since we have $\mathfrak{R}_{\epsilon_i}(X) \subseteq \mathfrak{R}_{\epsilon_j}(X)$ for any $\epsilon_i \leq \epsilon_j$, a VR filtration fulfilling the properties of Equation 6.3 can be performed by computing VR complexes at growing scales $\epsilon$.

During a filtration, we keep track of the scales at which a topological feature appears (birth) and disappears (death). For a given feature, e.g. the large cycle in Figure 6.2, this results in a tuple $(a, b)$ where $a$ represents the value of $\epsilon$ where the feature was created, and $b$ represents the value of $\epsilon$ when it got destroyed again. As a common choice to summarise the topological features that were extracted in such a filtration, we collect all such tuples in a *persistence diagram* where the $x$-axis represents the birth scale, and the $y$-axis reflects the death scale of the topological features. In Figure 6.3 we illustrate the persistence diagram that results from the VR filtration that was shown in Figure 6.2. We denote the persistent homology calculation of a VR complex (at multiple scales) with $\mathrm{PH}(\mathfrak{R}(X))$. As a result it returns a tuple $(\{\mathcal{D}_0, \mathcal{D}_1, \ldots\}, \{\pi_0, \pi_1, \ldots\})$. The first component contains a list of persistence diagrams, where $\mathcal{D}_d$ contains the persistence tuples of the $d$-dimensional topological features. The second component contains persistence pairings $\pi_d$ for dimension $d \in \{0, 1, \ldots\}$. For each tuple $(a, b)$ of a persistence diagram $\mathcal{D}_d$, the persistence pairing $\pi_d$ collects the indices $i, j$ of the simplices $\sigma_i, \sigma_j \in \mathfrak{R}(X)$ that triggered the birth and death events of the feature that the tuple represents.

We can compare two diagrams $\mathcal{D}, \mathcal{D}'$ by computing the bottleneck distance that is given by

$$\mathrm{d_b}\big(\mathcal{D}, \mathcal{D}'\big) := \inf_{\eta\colon \mathcal{D} \to \mathcal{D}'} \sup_{x \in \mathcal{D}} \|x - \eta(x)\|_\infty, \qquad (6.4)$$

where $\eta$ represents a bijection between the points in two diagrams and where $\|\cdot\|_\infty$ stands for the $\mathrm{L}_\infty$ norm. Finally, for ease of notation, we denote the set of persistence diagrams

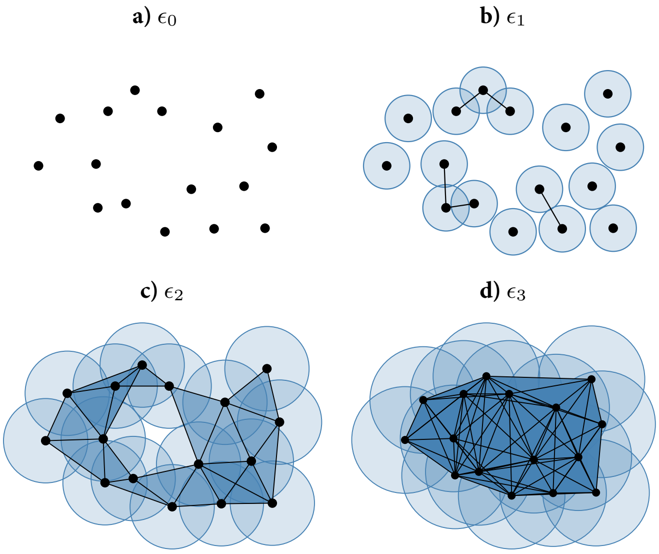


Figure 6.2: A visualisation of a Vietoris–Rips complex $\mathfrak{R}(X)$ of a point cloud $X$, shown for four increasing scales $\epsilon_0$ to $\epsilon_3$. By increasing the scale $\epsilon$, the connectivity also increases. For instance, around $\epsilon_1$ a cycle appears (captured via the $H_1$ homology group), whereas around $\epsilon_2$ the cycle feature disappears again. As a side note, for better readability we display balls of radius $\frac{1}{2}\epsilon$.

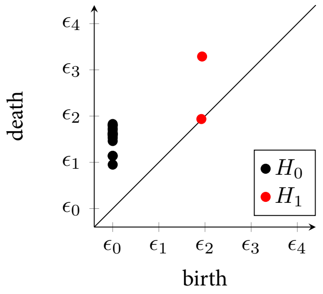


Figure 6.3: Persistence diagram of the Vietoris–Rips filtration shown in Figure 6.2. Here, we overlay the birth ($x$-axis) and death ($y$-axis) events of topological features for connected components (corresponding to the Betti number $\beta_0$ and the homology group $H_0$) as black dots, and cycles (corresponding to the Betti number $\beta_1$ and the homology group $H_1$) as red dots.

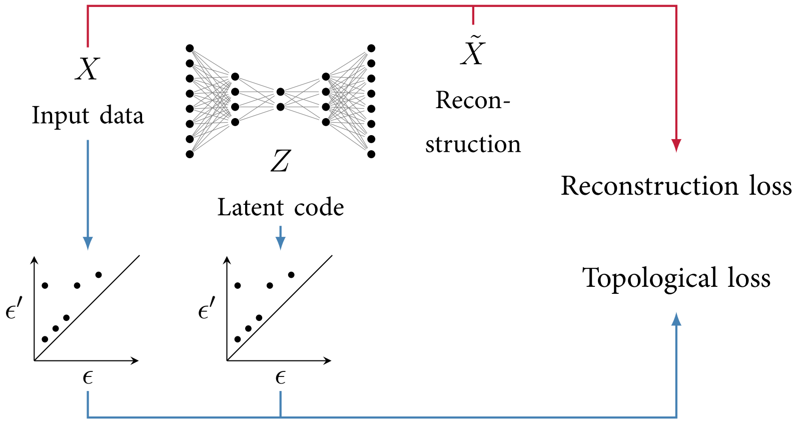


Figure 6.4: Overview of the topological autoencoder. A mini-batch $X$ is passed through an autoencoder, i.e., a deep neural network with a bottleneck, in order to reconstruct the input data as closely as possible with $\tilde{X}$. On top of the standard reconstruction loss, we compute a topological loss term that tracks how well the topological features of the data space and the latent space are aligned (calculated on the level of mini-batches). The idea behind this loss term is to act as a regulariser, to constrain the encoder such that the topology of the data space is preserved in the lower-dimensional latent space.

resulting from the persistent homology (PH) calculation of the point cloud $X$ as $\mathcal{D}^X$. Having introduced the preliminaries, we next introduce our new method.

## 6.4 A TOPOLOGY-PRESERVING AUTOENCODER

We propose a novel method for learning representations with autoencoders that preserve the topological features of the input space in the latent embeddings. In Figure 6.4, we give an initial overview of the method, while in the subsequent paragraphs, we provide more details about the individual steps and how they were implemented.

VIETORIS–RIPS COMPLEX CALCULATION Let $(\mathcal{S}, \mathrm{d})$ be a finite metric space. Considering $\mathcal{S}$ as a point cloud, and using the metric $\mathrm{d}\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, we compute the pairwise distance matrix of all points in $\mathcal{S}$ and denote it with $\mathbf{A}^{\mathcal{S}}$, where $\left(\mathbf{A}^{\mathcal{S}}\right)_{ij} = \mathrm{d}(s_i, s_j)$ for $s_i, s_j \in \mathcal{S}$. $\mathbf{A}^{\mathcal{S}}$ is sufficient for calculating the Vietoris–Rips complex (VR), and thanks to Observation 1 is also sufficient for the PH calculation. Even though the Euclidean metric is frequently used, persistent homology (as well as our method) is more general and can even be used with measures of similarity that do not fulfil the metric properties [206].

**Observation 1.** *For a finite metric space $\mathcal{S}$, the persistent homology of its Vietoris–Rips filtration is fully determined by the set of Vietoris–Rips complexes constructed at the finite set of scales $E = \{\epsilon \mid \epsilon = \mathrm{d}(s_i, s_j),\ \forall s_i, s_j \in \mathcal{S}\}$. Since $\mathcal{S}$ is finite, we have $x \in E \implies x \in \mathbf{A}^{\mathcal{S}}$.*

*Proof.* During the Vietoris–Rips filtration, a new topological feature can only be created or destroyed at a scale $\epsilon$ if the connectivity of the corresponding VR complex $\mathfrak{R}_\epsilon$ is different from $\mathfrak{R}_{\epsilon-\delta}$ such that $\mathfrak{R}_{\epsilon-\delta} \subset \mathfrak{R}_\epsilon$ for some $\delta$ with $0 < \delta \ll 1$. This implies that the birth and death of a feature that appears as a tuple in the resulting persistence diagram can be mapped to a threshold $\epsilon$ (that led to the birth or death) and a non-empty set of simplices $A$ for which holds that $A \subseteq \mathfrak{R}_\epsilon$, $A \nsubseteq \mathfrak{R}_{\epsilon-\delta}$, and $\mathfrak{R}_\epsilon = \mathfrak{R}_{\epsilon-\delta} \cup A$.

Following Definition 14, $\mathfrak{R}_\epsilon$ contains all simplices $\sigma \in \mathcal{S}$ for which $\epsilon$ is an upper bound of the diameter of $\sigma$. Since the diameter is determined by the supremum of the pairwise distances of the vertices in $\sigma$, a "new" simplex can only ever be introduced at scales $\epsilon$ that coincide with a distance $\mathrm{d}(s_i, s_j) \in \mathcal{S}$ that is observed in the finite metric space $\mathcal{S}$.

■

Since the VR complex is a clique complex, which means that it is fully determined by its vertices and edges [223], the PH calculation using a VR filtration can be interpreted as a selection of edges that are deemed topologically "relevant", i.e., responsible for the birth or death of a topological feature. For our application, this detail matters, which is why we track the edge indices corresponding to birth and death events in the persistence pairings $\pi_d$.

### 6.4.1 TOPOLOGICAL AUTOENCODER

Let $\mathcal{X}$ be a set, the data space. We consider the point cloud $X \subseteq \mathcal{X}$ to be a mini-batch of size $m$. Next, we define an autoencoder to be the composition of two mappings $h \circ g$. $g\colon \mathcal{X} \to \mathcal{Z}$ represents the *encoder* that maps input data to a latent space; $h\colon \mathcal{Z} \to \mathcal{X}$ represents the *decoder* that maps back from the latent space to the data space. For a mini-batch $X$, we denote the corresponding latent code as $Z = g(X)$. Figure 6.4 indicates that during a forward pass of our autoencoder architecture, the persistent homology is computed both for the mini-batch in the data space and the latent code, resulting in persistence diagrams and pairings in both spaces: $\left(\mathcal{D}^X, \pi^X\right) := \mathrm{PH}(\mathfrak{R}(X))$, and $\left(\mathcal{D}^Z, \pi^Z\right) := \mathrm{PH}(\mathfrak{R}(Z))$.

By subsetting the distance matrix $\mathbf{A}^X$ with the edge indices provided by the persistence pairings $\pi^X$, we can recover the values of the persistence diagrams. We denote this by $\mathcal{D}^X \simeq \mathbf{A}^X[\pi^X]$, to indicate that both the diagram and the subsetted distance matrix essentially contain the same information. Moreover, for ease of notation we treat $\mathbf{A}^X[\pi^X]$ as a vector in $\mathbb{R}^{|\pi^X|}$. We construct a topological regularisation term $\mathcal{L}_t := \mathcal{L}_t\left(\mathbf{A}^X, \mathbf{A}^Z, \pi^X, \pi^Z\right)$

by comparing both diagrams $\mathcal{D}^X$ and $\mathcal{D}^Z$ and add it (weighted by some parameter $\lambda \in \mathbb{R}$) to the reconstruction loss term $\mathcal{L}_r$ of the autoencoder to arrive at our overall loss:

$$\mathcal{L} = \mathcal{L}_r \left( X, (h \circ g)(X) \right) + \lambda \, \mathcal{L}_t \, . \tag{6.5}$$

Next, we consider the differentiability of the persistence diagram entries in order to construct the topological loss term $\mathcal{L}_t$. Conventional approaches to compare two persistence diagrams, for instance the bottleneck distance (see Equation 6.4), are too general in that they allow for the comparison of two unrelated diagrams. In our case, however, we have a one-to-one correspondence of individual data points between $X$ and $Z$ that we will subsequently exploit.

Differentiable persistence diagrams Our PH calculation can be interpreted as a selection of topologically relevant *distances* from the pairwise distance matrix. Moreover, each entry of our persistence diagrams corresponds to a distance between two data points. As a common assumption in the persistent homology literature [88, 159], we assume that the encountered distances are unique. This implies that for each entry $(a, b)$ of a diagram, its infinitesimal neighbourhood contains only the point $(a, b)$. While this assumption could in principle be violated, in practice, uniqueness could be achieved via small perturbations of the data. Given such a fixed[4] persistence pairing, and assuming a differentiable distance function d, the entries of the persistence diagram of the latent space, $\mathcal{D}^Z$, are also differentiable with regard to the parameters of the encoder $g$. This implies the existence of the derivative of a loss function that utilises the persistence diagrams, which permits us to obtain gradients for backpropagation.

The topological loss term Having established that we can use persistence diagram entries in a differentiable loss term, we now consider how to best construct a loss term that allows us to preserve the structure of the data space (in terms of topological features) in the latent encodings. A straightforward solution would be to directly compare the selected distances of the two spaces. However, such an approach would not be informative as it merely compared the values of diagram entries without a pairwise correspondence between data points and latent codes.

A more elaborate approach would be to enforce similarity of the those diagram entries that correspond to the same VR complex edges in both spaces. However, the intersection of the two persistence pairings would include very few edges upon initialisation of the model

---

[4] This means that small perturbations of the data do not affect which edges of the VR complex are selected in the PH calculation.

parameters, leading to unstable training due to uninformative gradients and biased estimates of the alignment of the topological features between the two spaces. We overcome this challenges by considering the *union* of the selected edges in the data space as well as the latent space. Our topological loss $\mathcal{L}_t$ comprises two directed components, where in each component the topological features (in terms of the persistence pairings) are kept fixed for one space:

$$\mathcal{L}_t := \mathcal{L}_{\mathcal{X}\to\mathcal{Z}} + \mathcal{L}_{\mathcal{Z}\to\mathcal{X}}, \tag{6.6}$$

where

$$\mathcal{L}_{\mathcal{X}\to\mathcal{Z}} := \frac{1}{2}\left\|\mathbf{A}^X\left[\pi^X\right] - \mathbf{A}^Z\left[\pi^X\right]\right\|^2 \tag{6.7}$$

and

$$\mathcal{L}_{\mathcal{Z}\to\mathcal{X}} := \frac{1}{2}\left\|\mathbf{A}^Z\left[\pi^Z\right] - \mathbf{A}^X\left[\pi^Z\right]\right\|^2. \tag{6.8}$$

This formulation of the loss allows us to account for at least $|X|$ topologically relevant distances. In case two spaces $\mathcal{X}, \mathcal{Z}$ were perfectly aligned, $\mathcal{L}_{\mathcal{X}\to\mathcal{Z}} = \mathcal{L}_{\mathcal{Z}\to\mathcal{X}} = 0$, since the pairings as well as the selected distances coincide. However, the converse is not necessarily true. Even though $\mathcal{L}_t = 0$ implies that the compared distances are identical, the underlying pairing still could differ. Therefore, since $\mathcal{L}_t$ violates the identity of indiscernibles, it does not meet the criteria for a metric. Having specified our topological loss term, we next showcase how we can calculate its gradients.

GRADIENT CALCULATION For an encoder $g\colon \mathcal{X} \to \mathcal{Z}$, let a vector $\boldsymbol{\theta} \in \mathbb{R}^k$ denote its parameters. Letting $\boldsymbol{\rho} = \left(\mathbf{A}^X\left[\pi^X\right] - \mathbf{A}^Z\left[\pi^X\right]\right)$, we have

$$\frac{\partial}{\partial\boldsymbol{\theta}}\mathcal{L}_{\mathcal{X}\to\mathcal{Z}} = \frac{\partial}{\partial\boldsymbol{\theta}}\left(\frac{1}{2}\left\|\mathbf{A}^X\left[\pi^X\right] - \mathbf{A}^Z\left[\pi^X\right]\right\|^2\right) \tag{6.9}$$

$$= -\boldsymbol{\rho}^\top\left(\frac{\partial\mathbf{A}^Z\left[\pi^X\right]}{\partial\boldsymbol{\theta}}\right) = -\boldsymbol{\rho}^\top\mathbf{J}, \tag{6.10}$$

where $\mathbf{J} \in \mathbb{R}^{|\pi^X|\times k}$ represents the Jacobian with entries $\mathbf{J}_{ij} = \frac{\partial\mathbf{A}^Z[\pi^X]_i}{\theta_j}$, $|\pi^X|$ represents the cardinality of $\pi^X$, and $\mathbf{A}^Z[\pi^X]_i$ denotes the $i^{\text{th}}$ entry of the vector of distances selected from the distance matrix of the latent space using the pairing $\pi^X$. Analogously, this can be derived for $\mathcal{L}_{\mathcal{Z}\to\mathcal{X}}$. In both cases, derivatives of entries of $\mathbf{A}^X$ with respect to $\boldsymbol{\theta}$ vanish since the distances in the data space do not depend on the encoder (in contrast to the distances in the latent space). Even though the persistence diagrams change during the training process in a non-differentiable way, for a given loss calculation at an individual update step, the diagrams are robust to infinitesimal perturbations [42], keeping the gradients of our loss term well-defined.

### 6.4.2 STABILITY

Even though persistence diagrams are stable with regards to perturbations in the data, in this section we show that persistence diagrams are also robust under subsampling of the underlying data, which in turn justifies the scalable computation of our loss on the mini-batch level.

**Theorem 1.** *For a point cloud $X$ of size $n$, let $X^{(m)} \subseteq X$ denote a subset of $X$ of size $m$ that we refer to as* subsample*. Comparing the persistence diagrams of $X$ and $X^{(m)}$, we have the following bound:*

$$\mathbb{P}\left(\mathrm{d_b}\left(\mathcal{D}^X, \mathcal{D}^{X^{(m)}}\right) > \epsilon\right) \leq \mathbb{P}\left(\mathrm{d_H}\left(X, X^{(m)}\right) > 2\epsilon\right), \tag{6.11}$$

*where $\mathrm{d_b}(\cdot,\cdot)$ represents the bottleneck distance (see Equation 6.4) between the two persistence diagrams and $\mathrm{d_H}(\cdot,\cdot)$ denotes the Hausdorff distance between the point cloud and its subsample, i.e.,*

$$\begin{aligned} \mathrm{d_H}(X, Y) := \max\{ & \sup_{x\in X} \inf_{y\in Y} \mathrm{d}(x, y), \\ & \sup_{y\in Y} \inf_{x\in X} \mathrm{d}(x, y)\} \end{aligned} \tag{6.12}$$

*using a base distance $\mathrm{d}(x, y)$, e.g. the Euclidean distance.*

*Proof.* Chazal et al. [30] proved the stability of the calculation of persistent homology in finite metric spaces. Specifically, for two metric spaces $X$ and $Y$, we have

$$\mathrm{d_b}\left(\mathcal{D}^X, \mathcal{D}^Y\right) \leq 2\,\mathrm{d_{GH}}(X, Y), \tag{6.13}$$

where $\mathrm{d_{GH}}(\cdot,\cdot)$ represents the Gromov-Hausdorff distance, which is defined as the lower bound of the Hausdorff distance evaluated over all isometric embeddings of the spaces $X$ and $Y$ [23]. However, $\mathrm{d_{GH}}$ is difficult to compute. Furthermore, since $Y = X^{(m)}$, both spaces $X$ and $Y$ share the same metric, in our case. By definition, we have $\mathrm{d_{GH}}(X, Y) \leq \mathrm{d_H}(X, Y)$. Together with Equation 6.13 we can bound the bottleneck distance of the diagrams directly with the Hausdorff distance

$$\mathrm{d_b}\left(\mathcal{D}^X, \mathcal{D}^Y\right) \leq 2\,\mathrm{d_H}(X, Y), \tag{6.14}$$

from which the claim in Equation 6.11 follows upon taking probabilities of both the left and right hand side of Equation 6.14. ■

Theorem 1 relates the comparison of persistence diagrams to the Hausdorff distance, a distance measure on the underlying point clouds. In the following theorem, we further investigate the Hausdorff distance between a point cloud and its subsample to provide an upper bound of its expectation.

**Theorem 2.** *Given a metric* $\mathrm{d}$*, let* $\mathbf{A} \in \mathbb{R}^{n \times m}$ *denote the pairwise distance matrix between a point cloud* $X$ *and its subsample* $X^{(m)}$ *of size* $m$*.* $\mathbf{A}$ *is sorted such that the first* $m$ *rows coincide with the* $m \times m$ *distance matrix of* $X^{(m)}$*. We assume that elements* $a_{ij}$ *for* $i > m$ *are randomly drawn samples of a distribution of distances* $F_D$ *with non-negative support. Further, the row minima* $\delta_i$ *of rows* $i > m$ *follow the distribution* $F_\Delta$*. Let* $Z := \max_{1 \leq i \leq n} \delta_i$ *follow a distribution* $F_Z$*. Then, the with respect to* $F_Z$ *expected Hausdorff distance between* $X$ *and its subsample* $X^{(m)}$ *for* $m < n$ *can be bounded with*

$$\mathbb{E}\Big[\mathrm{d}_{\mathrm{H}}(X, X^{(m)})\Big] = \mathbb{E}_{Z \sim F_Z}[Z] \tag{6.15}$$

$$\leq \int_0^{+\infty} \left(1 - F_\Delta(z)^{n-m}\right) \mathrm{d}z, \tag{6.16}$$

*where*

$$F_\Delta(z) \approx -\sum_{k=1}^{m} \binom{m}{k} \left(-F_D(z)\right)^{m-k}. \tag{6.17}$$

We prepare the subsequent proof of this claim by first highlighting two observations.

**Observation 2.** *Due to the relation* $X^{(m)} \subseteq X$*, it follows that*

$$\sup_{x' \in X^{(m)}} \inf_{x \in X} \mathrm{d}\big(x, x'\big) = 0. \tag{6.18}$$

*Therefore, the computation of the Hausdorff distance can be simplified such that*

$$\mathrm{d}_{\mathrm{H}}\Big(X, X^{(m)}\Big) := \sup_{x \in X} \inf_{x' \in X^m} \mathrm{d}\big(x, x'\big). \tag{6.19}$$

**Observation 3.** *As the point clouds of consideration represent finite sets, the infimum and supremum operations coincide with the minimum and maximum, and for the sake of better readability can be replaced, accordingly.*

Building on these observations, we can decompose the calculation of $\mathrm{d_H}\big(X, X^{(m)}\big)$ into the following three steps:

1. Given a base distance d, we compute the $n \times m$ distance matrix **A** between the point cloud $X$ and its subsample $X^{(m)}$.
2. For each sample in $X$, we determine the minimal distance to the $m$ points of $X^{(m)}$ via the extraction of the row minimum of **A** in order to collect all minimal distances in a vector $\boldsymbol{\delta} \in \mathbb{R}^n$.
3. We retrieve the desired result as the maximal entry of $\boldsymbol{\delta}$, $\max(\boldsymbol{\delta}) = \mathrm{d_H}\big(X, X^{(m)}\big)$.

*Proof.* Leveraging Observations 2 and 3, we arrive at the following simplification for the Hausdorff distance:

$$\mathrm{d_H}\left(X, X^{(m)}\right) := \max_{i, 1 \leq i \leq n} \left( \min_{j, 1 \leq j \leq m} (a_{ij}) \right). \tag{6.20}$$

Since diagonal entries $a_{ii} = 0$ for $1 \leq i \leq m$, the minimal distances of the first $m$ rows must be 0. Therefore, the outer bracket maximum in Equation 6.20 will be determined by the last $n - m$ row minima $\{\delta_i \mid m < i \leq n\}$ of **A** where $\delta_i := \min\limits_{1 \leq j \leq m} a_{ij}$. Those exact minima were assumed to follow a distribution $F_\Delta(y)$ for which we have

$$\begin{aligned}
F_\Delta(y) &= \mathbb{P}(\delta_i \leq y) = 1 - \mathbb{P}(\delta_i > y) && (6.21)\\
&= 1 - \mathbb{P}\left(\min_{1 \leq j \leq m} a_{ij} > y\right) && (6.22)\\
&= 1 - \mathbb{P}\left(\bigcap_j a_{ij} > y\right) && (6.23)\\
&\approx 1 - (1 - F_D(y))^m && (6.24)\\
&= -\sum_{k=1}^{m} \binom{m}{k} (-F_D(y))^{m-k}, && (6.25)
\end{aligned}$$

where Equation 6.24 is approximated by assuming independent sampling from $F_D$. Next, considering $Z := \max_{1 \leq i \leq n} \delta_i$, we derive how $Z$ is distributed:

$$\begin{aligned}
F_Z(z) = \mathbb{P}(Z \leq z) &= \mathbb{P}\left(\max_{m < i \leq n} \delta_i \leq z\right) && (6.26)\\
&= \mathbb{P}\left(\bigcap_{m < i \leq n} \delta_i \leq z\right) && (6.27)
\end{aligned}$$

We continue by approximating $Z$ with $Z'$ where we impose that the row minima $\delta_i$ are sampled i.i.d. from $F_\Delta$. In practice, the entries of the last $n-m$ rows of **A** are not independent, but carry information about each value, for instance via the metric triangular inequality. If we nevertheless assume i.i.d., we have:

$$F_{Z'}(z) = F_\Delta(z)^{n-m}. \tag{6.28}$$

As $Z'$ has non-negative support, the expectation evaluates to:

$$\mathbb{E}_{Z' \sim F_{Z'}}\left[Z'\right] = \int_0^{+\infty} (1 - F_{Z'}(z))\,\mathrm{d}z \tag{6.29}$$

$$= \int_0^{+\infty} \left(1 - F_\Delta(z)^{n-m}\right)\mathrm{d}z \tag{6.30}$$

Next, to understand how the assumption of i.i.d. sampled distance values (and therefore row minima) affects $Z'$ compared to $Z$, we consider a set of points of the point cloud and the metric d. Let $x_1, x_2, x_3 \in X$ be three points of the finite metric space $(X, \mathrm{d})$. Furthermore, we abbreviate $\mathrm{d}_{ij} = \mathrm{d}(x_i, x_j)$. We further assume that $\mathrm{d}_{12}$ happens to be a row minimum $\delta_i$ as defined above. The triangular inequality tells us that $\mathrm{d}_{23} - \mathrm{d}_{13} \leq \mathrm{d}_{12} \leq \mathrm{d}_{23} + \mathrm{d}_{13}$. Therefore, $\mathrm{d}_{12}$ does not represent a random sample from the empirical distribution $F_\Delta$, but instead a sample of the constrained distribution $F_\Delta |\, \mathrm{d}_{23} - \mathrm{d}_{13} \leq z \leq \mathrm{d}_{23} + \mathrm{d}_{13}$[5], where both tails of the distribution $F_\Delta$ are cropped away. Since we construct $Z'$ by assuming that the row minima $\delta_i$ were sampled independently, the row minima underlying $Z'$ would keep the tails of the distribution $F_\Delta$ which implies that the empirical (sample) maximum of the row minima are overestimated in $Z'$. Hence, the expectation of the maximal $\delta_i$, i.e., $\mathbb{E}[Z']$, is (with high probability) an upper bound for the actual $\mathbb{E}[Z]$, and therefore for $\mathbb{E}\left[\mathrm{d}_\mathrm{H}(X, X^{(m)})\right]$.

■

In Moor et al. [141], Section A.3 of the Supplementary materials, we further investigated empirical convergence rates of the Hausdorff distance between a point cloud $X$ and its subsample $X^{(m)}$. Finally, we conclude that subsampling a point cloud is stable and a suitable approach for approximating the topological features available in the full point cloud.

---

[5] For the sake of the argument, we denote only one such constraint, however the metric induces numerous constraints of this kind.

### 6.4.3 SCALABILITY AND RUNTIME

We briefly discuss scalability considerations. For 0-dimensional features, i.e., connected components, we only need to consider *edges* of the VR complex that lead to death events in the persistence diagram, and are captured in $\pi_0$. The computation of these edges is efficient in that the worst-case runtime complexity is $\mathcal{O}\left(m^2 \cdot \alpha\left(m^2\right)\right)$, for a mini-batch size $m$ and where $\alpha(\cdot)$ represents the inverse Ackermann function which grows extremely slowly [43, Chapter 22]. For 1-dimensional features, i.e., cycles, we can obtain edges by pairing triangles with its edge that has the largest weight (in our case, simply the largest edge). This method can be generalised to higher dimensions by always pairing the corresponding simplex with an edge. However, in preliminary tests the inclusion of 1-dimensional features was not beneficial and only increased the runtime. Therefore, in our current setup we focused on 0-dimensional features.

## 6.5 RELATED WORK

Topological data analysis and specifically persistent homology (PH) have gained attention across multiple areas of machine learning research and applications. Conventionally, PH was frequently used to analyse topological features of a dataset in a post-hoc manner. Several works characterised the topological features of high-dimensional data as compared to embeddings [108, 154, 170, 171, 215]. PH has also been used to analyse the training as well the decision boundary of deep neural networks [77, 161, 172]. Our work differs from these previous publications in that our differentiable regularisation term constrains the model during training in order to preserve topological features. Furthermore, several publications have explored how to integrate topological features into classifiers in order to improve performance. For instance, Hofer et al. [89] proposed a layer for neural networks to learn projections of a persistence diagram. Several strategies to vectorise persistence diagrams have been proposed, making it easier to include fixed-dimensional topological feature representations in classifiers [1, 26, 27]. These approaches, however, treat a persistence diagram as a fixed object, without the possibility of adjusting the input data in order to retrieve desired topological features in the diagrams. Such adjustments were only recently introduced. For instance, Poulenard et al. [159] optimised real-valued functions using their topology. While this can be seen as the first method for aligning persistence diagram by means of changing the input data, the method is limited by certain restrictions such as that the connectivity of the data needs to be known, or that it requires scalar-valued functions. Working directly with distances using the VR complex, we side-step this issue. While Hofer et al. [88] also proposed a differentiable loss, their formulations aims at enforcing a single scale $\eta$ in latent encodings that are used in a downstream classification task. In contrast to enforcing a single scale, we learn latent codes that preserve topological features at multiple scales as available in the data space.

## 6.6 EXPERIMENTS

In our experiments, we consider the unsupervised task to learn low-dimensional embeddings from data that best preserves the their topological features.

### 6.6.1 EXPERIMENTAL SETUP

In the following, we introduce the considered datasets, baselines, the training procedure, as well as our evaluation strategy.

DATASETS First, we create a SPHERES dataset consisting of a set of $100$-spheres living in $101$-dimensional space. Specifically, ten smaller spheres (each sampled at $500$ points) are enclosed by one larger sphere (sampled at $5{,}000$ points). The small spheres were defined by a radius $r = 5$, whereas each sphere was translated by a vector of Gaussian noise that per sphere was sampled once from the distribution $\mathcal{N}\left(\mathbf{0}, \mathbf{I}\frac{10}{\sqrt{d}}\right)$ with $d = 101$. To make this dataset topologically more interesting, we added a larger sphere that enclosed the smaller spheres with a radius $5r$. As for real-world datasets, we considered three image datasets, FASHION-MNIST, MNIST and CIFAR-10, which could be of particular interest to our approach since natural images have been shown to evolve along low-dimensional manifolds [156].

BASELINES AND TRAINING As for comparison partners, we include several techniques for dimensionality reduction. This includes uniform manifold approximation and projection (UMAP) [135], t-distributed stochastic neighbour embedding (t-SNE) [132], Isomap [194], principal component analysis (PCA), and classical autoencoders (AEs). By applying our topological regularisation term to the same autoencoder architecture, we create a topological autoencoder (TopoAE).

For maximal interpretability and comparability, we restricted each method to a two-dimensional latent space. If available, we used existing splits of the data and otherwise split the data into a 90% training set and 10% testing set. On top, we held out 15% of the training set as a validation set which was used to tune the hyperparameters. By dividing the topological loss term by the batch size $m$, we disentangled the regularisation strength $\lambda$ from the batch size. All included neural networks employed batch normalisation [96] and were fitted using ADAM [112]. Due to t-SNE not being intended for the application to unseen data, we evaluate this baseline only on the training split of the data. Due to significant scaling issues with Isomap we were not able to perform a hyperparameter search for this method on the real-world image datasets. Therefore, we included this approach only in the analysis of the synthetic dataset. For further details regarding the model architectures and hyperparameter tuning, please refer to [141, Section A.6].

EVALUATION We measure the quality of latent embeddings in the following three ways. We consider i) visualisations of the low-dimensional representations, ii) quality metrics for dimensionality reduction, and iii) the reconstruction error (Data MSE) which is computed between the input and reconstructed data, assuming a method could return reconstructions[6].

As for evaluation metrics contained in item ii), we investigate a battery of non-linear dimensionality reduction metrics [72] that can be obtained when comparing the data space with the latent space. This includes the

1. root mean square error ($\ell$-RMSE) between the distance matrices of the data space and the latent space,
2. mean relative rank error ($\ell$-MRRE), which quantifies changes in ranks of the distances when comparing the input space with the latent space [120],
3. trustworthiness ($\ell$-Trust) [202], a measure that which measures how well neighbourhoods (in terms of $k$ nearest neighbours) are preserved when moving from the *data* space to the latent space, and
4. continuity ($\ell$-Cont) [202], which analogously to $\ell$-Trust measures to which degree neighbourhood relations are preserved when moving from the *latent* space to the data space.

All of the above measures are prefixed with an $\ell$ to emphasise that they compare the input space with the latent space, i.e., reconstructions that are obtained by mapping back from the latent space to the data space are *not* taken into account here. In addition, we compute a Kullback–Leibler divergence of the distributions of densities compared between the input space and the latent space. Inspired by the *distance to a measure* density estimator [29, 31], for a point cloud $X$, we calculate the density of a point $x \in X$ with

$$\mathrm{f}_{\sigma}^{X}(x) := \sum_{y \in X} \exp\left(-\sigma^{-1}\,\mathrm{dist}(x, y)^{2}\right), \tag{6.31}$$

---

[6] This was the case for PCA and all autoencoder-based models.

with $\sigma \in \mathbb{R}_{>0}$ indicating a length scale, and where for dist, we used the Euclidean distance, normalised to the range $[0, 1]$. We then evaluate the mismatch between the density distributions of two point clouds $X$ and $Z$ with

$$\mathrm{KL}_{\sigma} := \mathrm{KL}\left(\mathrm{f}_{\sigma}^{X} \parallel \mathrm{f}_{\sigma}^{Z}\right), \tag{6.32}$$

where it is desirable to recover a small divergence, indicating that the density estimates of the low-dimensional encodings in $Z$ are similar to the densities in $X$. Furthermore, to account for different scales of the data, we report several choices of $\sigma$, where we minimised $\mathrm{KL}_{0.1}$ as the objective of our hyperparameter search.

### 6.6.2 RESULTS

We organise this section such that quantitative results (Section 6.6.2) are followed by qualitative visualisations of the latent embeddings (Section 6.6.2).

#### QUANTITATIVE RESULTS

We report our quantitative results in Table 6.1. We find that TopoAE is able to preserve the data density at multiple scales (in terms of $\mathrm{KL}_{\sigma}$). Also, we observe competitive values for continuity ($\ell$-Cont) and the reconstruction error (Data MSE). The second measure reveals that our imposed topological constraints do not lead to large impairments of the reconstruction quality. Further classical measures (starting with $\ell$) favoured the baselines, in particular the *training* performance of the t-SNE baseline. Interestingly, in the following section we will observe how the classical measures can fail at detecting crucial structural information when the manifold underlying the data is actually known.

#### VISUALISING THE LATENT SPACE

Figure 6.5 illustrates the latent spaces obtained for the SPHERES dataset. We find that only our method, TopoAE, was able to appropriately capture the nested configuration of the sphere manifolds. In contrast, t-SNE, a baseline that on this dataset shows excellent results in terms of the classical metrics of dimensionality reduction, fails to preserve this nesting relationship by severing the outer sphere (shown in dark blue). Interestingly, we observe that the KL-divergence is well aligned with the visual assessment that our method best preserves the structure of the known manifolds underlying this dataset, which makes the KL term a measure of particular interest in datasets, where the manifolds are unknown. Notably, since baselines such as t-SNE and Isomap faired well with regard to several classical measures, the fact that these methods could not capture the global structure of the dataset (whereas the outer sphere accounted for half of the dataset!) suggests that the classical measures may not faithfully track the relevant structural information at hand. For the Spheres dataset, we provide additional results in Section A.3 of the Supplementary materials. This includes an ablation of TopoAE by considering a linear autoencoder with a single hidden layer as well as a comparison to the PHATE method [137], both displayed in Figure A.20.

The visualisations for Fashion-MNIST are shown in the left column of Figure 6.6. When comparing TopoAE with AE, its unregularised counterpart that focuses only on minimising the reconstruction error, we observe that TopoAE is additionally constrained to preserve structure leading to a more organised latent space, resulting in visually similar patterns as with UMAP, the one baseline which also takes a topological perspective on the data [135]. In addition, t-SNE shows a tendency to fragment a cluster of one class into several subgroups. As this type of artefact is commonly encountered with t-SNE, this pattern is likely not revealing interesting substructures of the underlying manifold. The center column of Figure 6.6 represent the embeddings for the MNIST dataset. Here, we observe that by means of pulling apart clusters of distinct classes, in the embeddings of the non-linear baselines, some spatial relationships between classes are lost, as compared to PCA or TopoAE. Finally, the right column of Figure 6.6 visualises the embeddings of CIFAR-10, where we observe that this dataset is hard to embed into two dimensions without supervision. Nevertheless, we observe that our method identified a linear structure potentially dividing the latent space. Since TopoAE is designed to preserve shapes and structures of the input space, this pattern could be indicative of the manifold structure underlying this dataset.

## 6.7 Discussion

In this chapter, we introduced topological autoencoders (TopoAEs), a novel deep learning method for learning low-dimensional representations of data that preserve multi-scale topological information, therefore revealing complex structures in otherwise intangible high-dimensional data spaces. We demonstrated that under weak assumptions, our topological regularisation term based on persistent homology can be integrated into an end-to-end differentiable model trained using backpropagation. Furthermore, we showed that the persistent homology of a dataset can be robustly approximated on the mini-batch level, indicating that our scalable loss term (that only requires samples from a mini-batch) is theoretically founded and actually tracks the topological features of the data space.

As for empirical results, we found that our method was uniquely capable of recovering complex relationships between samples from nested sphere manifolds in high dimensions.

| Data set | Method | $\mathrm{KL}_{0.01}$ | $\mathrm{KL}_{0.1}$ | $\mathrm{KL}_{1}$ | $\ell$-MRRE | $\ell$-Cont | $\ell$-Trust | $\ell$-RMSE | Data MSE |
|---|---|---|---|---|---|---|---|---|---|
| SPHERES | Isomap | 0.181 | **0.420** | **0.00881** | **0.246** | **0.790** | **0.676** | 10.4 | – |
| | PCA | 0.332 | 0.651 | 0.01530 | 0.294 | 0.747 | 0.626 | 11.8 | 0.9610 |
| | TSNE | **0.152** | 0.527 | 0.01271 | **0.217** | 0.773 | **0.679** | **8.1** | – |
| | UMAP | 0.157 | 0.613 | 0.01658 | 0.250 | 0.752 | 0.635 | **9.3** | – |
| | AE | 0.566 | 0.746 | 0.01664 | 0.349 | 0.607 | 0.588 | 13.3 | **0.8155** |
| | TopoAE | **0.085** | **0.326** | **0.00694** | 0.272 | **0.822** | 0.658 | 13.5 | **0.8681** |
| F-MNIST | PCA | **0.356** | **0.052** | **0.00069** | 0.057 | 0.968 | 0.917 | **9.1** | 0.1844 |
| | TSNE | 0.405 | 0.071 | 0.00198 | **0.020** | 0.967 | **0.974** | 41.3 | – |
| | UMAP | 0.424 | 0.065 | 0.00163 | 0.029 | **0.981** | 0.959 | **13.7** | – |
| | AE | 0.478 | 0.068 | 0.00125 | **0.026** | 0.968 | **0.974** | 20.7 | **0.1020** |
| | TopoAE | **0.392** | **0.054** | **0.00100** | 0.032 | **0.980** | 0.956 | 20.5 | **0.1207** |
| MNIST | PCA | 0.389 | 0.163 | 0.00160 | 0.166 | 0.901 | 0.745 | **13.2** | 0.2227 |
| | TSNE | **0.277** | **0.133** | 0.00214 | **0.040** | 0.921 | **0.946** | 22.9 | – |
| | UMAP | **0.321** | 0.146 | 0.00234 | **0.051** | **0.940** | **0.938** | **14.6** | – |
| | AE | 0.620 | 0.155 | **0.00156** | 0.058 | 0.913 | 0.937 | 18.2 | **0.1373** |
| | TopoAE | 0.341 | **0.110** | **0.00114** | 0.056 | **0.932** | 0.928 | 19.6 | **0.1388** |
| CIFAR | PCA | **0.591** | **0.020** | **0.00023** | 0.119 | **0.931** | 0.821 | **17.7** | 0.1482 |
| | TSNE | 0.627 | 0.030 | 0.00073 | **0.103** | 0.903 | **0.863** | **25.6** | – |
| | UMAP | 0.617 | 0.026 | 0.00050 | 0.127 | 0.920 | 0.817 | 33.6 | – |
| | AE | 0.668 | 0.035 | 0.00062 | 0.132 | 0.851 | **0.864** | 36.3 | **0.1403** |
| | TopoAE | **0.556** | **0.019** | **0.00031** | **0.108** | **0.927** | 0.845 | 37.9 | **0.1398** |

Table 6.1: Quantitative evaluation of the latent embeddings in terms of how well they preserve the structure and neighbourhoods of the high-dimensional input data (see Section 6.6.1 for a description of the evaluation metrics). Hyperparameters were tuned by minimising $\mathrm{KL}_{0.1}$. For each measure, shown in a separate column, we display the winner both underlined and bold, whereas the runner-up is shown in bold. For a more detailed version of this table (including variances and more scales), please refer to [141, Table A.2].

This result is interesting also from a manifold learning perspective, where it remains challenging for models to seamlessly cope with multiple manifolds in the domain [52]. On real-world image data, we found that our topological loss results in competitive performance as measured with several quality metrics (for instance the preservation of densities at multiple scales), while not impairing the models ability to reconstruct data. Both in synthetic datasets, where the to-be-learned manifold is known, as well as in real-world datasets, TopoAE learned faithful and interesting representations. Compared to several non-linear comparison partners which focus on local scales, our method did not pull apart clusters of distinct classes, but organised them in entangled structures, that plausibly reveal a more realistic depiction of the underlying manifold (as could be validated in the case of the SPHERES dataset).

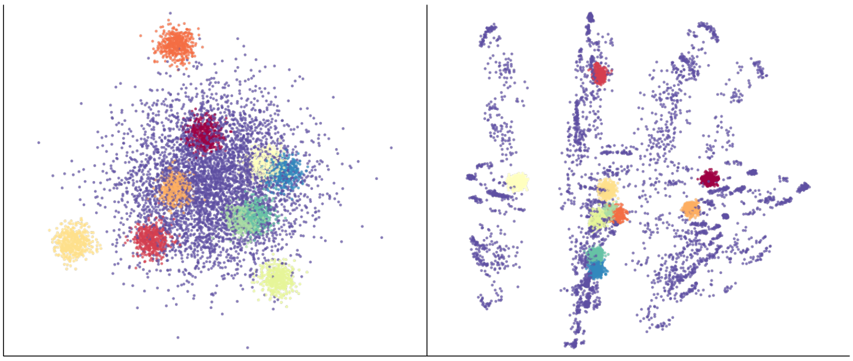
To keep our experimental setup fair for non-topological methods, we intentionally omitted evaluation measures that specifically compare the topological features between the data and latent spaces. However, in an auxiliary analysis, we empirically confirmed that our method indeed preserves topological features in the latent space (see [141, Section A.10]).

FUTURE WORK We formulated our topological loss in a highly generic manner. In essence, we merely require data objects (i.e., encoded as tensors that can be parsed by standard neural networks) and distances between these objects. Therefore, this method can be integrated into architectures different from the ones showcased in this chapter. To give a few examples, this constraint may be applied to variational models ([141, Figure A.3]), or can even make simpler methods, such as PCA, topology-aware ([141, Figure A.6]). Nevertheless, deploying our loss term to settings with more involved architectures remains an exciting route to be explored in future work. Furthermore, in our current formulation the loss term largely depends on having chosen an appropriate distance function in the data space. Following up on this, we found that the euclidean distance performs surprisingly well on image datasets when compared to perceptually inspired distances [140].

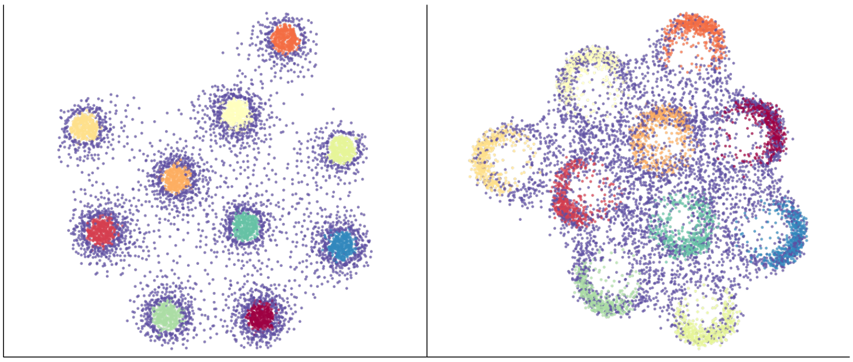
As one limitation to our method, we currently focus on 0-dimensional features, while including higher-dimensional features could lead to scalability issues if the mini-batch size becomes large. However, in our current setup, we observed that for smaller batch sizes the runtime even *increases*, i.e., the efficiency of larger batches still dominated the increased cost of computing larger VR complexes. Effectively scaling to higher-dimensional features could be achieved with approximations to the persistent homology calculation [41], or by means of parallelism [123] and GPU acceleration [219].

Finally, we envision that topology-aware models will offer deep insights into challenging and complex biomedical datasets by means of adding a structural perspective that is typically neglected by conventional methods. Initial works on topology-aware machine learning (e.g., topological autoencoders [141], topological graph neural networks [92], or topological attention models [217]) have brought forth a rich toolbox of methods that will be exciting to employ in biomedical datasets in order to uncover and fully leverage the shape and structure of these data.

**a)** PCA **b)** Isomap



**c)** t-SNE **d)** UMAP



**e)** AE **f)** TopoAE

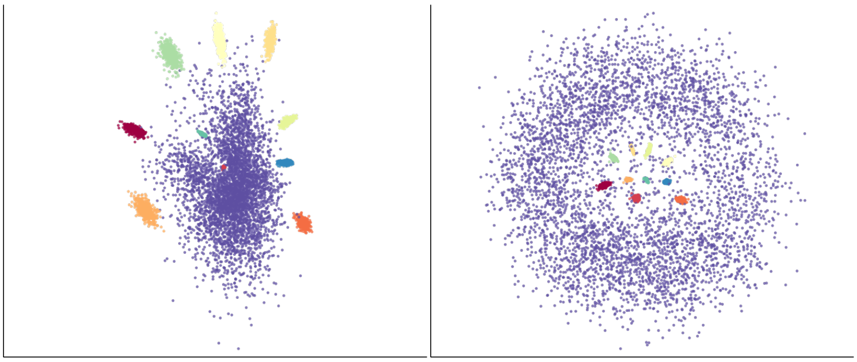


Figure 6.5: Visualisations of the two-dimensional latent embeddings of the SPHERES dataset. We observe that only our method, TopoAE was able to accurately capture the nested configuration of the sphere manifolds. In comparison, t-SNE, a method that performs well on this dataset in terms of classical dimensionality reduction metrics, tears the enclosing sphere apart.

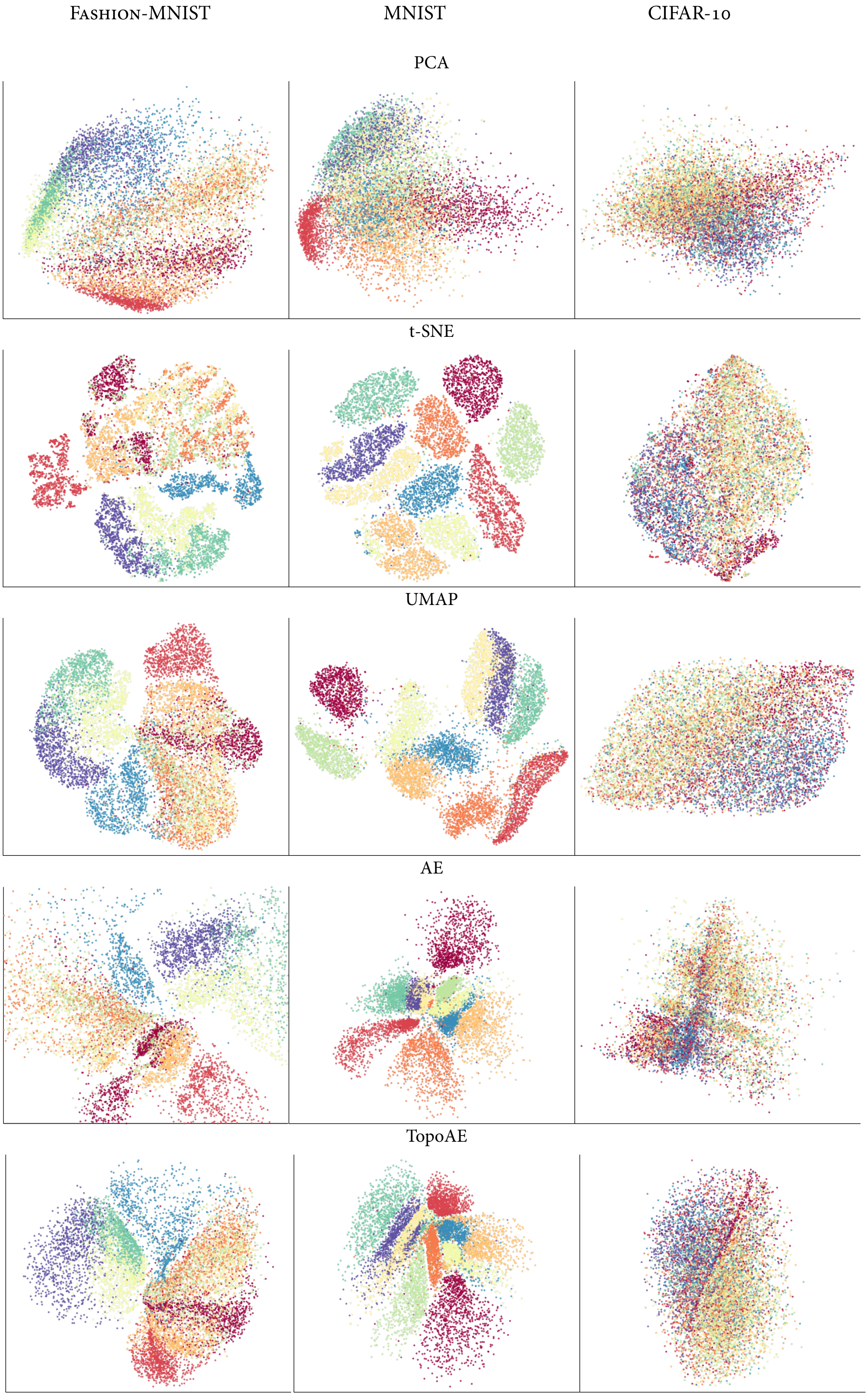


Figure 6.6: Visualisations of the two-dimensional latent embeddings for the datasets Fashion-MNIST (left column), MNIST (center column), and CIFAR-10 (right column). The corresponding method is indicated above each row.

# 7 CONCLUSION

At the beginning of this dissertation, we outlined a set of challenges that arise when considering clinical prediction problems in a data-driven way using machine learning (ML). This includes

- DATASETS: the lack of accessible, and annotated large datasets,
- TASKS: the difficulty to identify a meaningful prediction task which can plausibly produce clinical value,
- LABELS: the lack of ground-truth labels that can be used for training a ML model,
- VALIDITY: the challenge to learn models that generalise to unseen data in spite of a variety of distribution shifts at hand, and
- MISSINGNESS: the problem of accounting for missing data, as well the sampling information its conveys.

Over the course of this thesis, we have encountered these problems, and have proposed solutions to mitigate them. In the first part, which was focused on applications, we developed clinical prediction models for the classification of patient time series, specifically in order to detect sepsis. In a second, more method-focused part, motivated by the learning of robust and uncertainty-aware representations during Part I, we consider model-internal states (as opposed to predictions that the models output), i.e., we focus on learned representations that lead to beneficial downstream performance, or that more faithfully reflect the input data.

This chapter serves for collecting and summarising the findings of the individual chapters of this dissertation. Finally, we conclude this thesis by envisioning future directions that build on the foundations laid out by this work.

## 7.1 PART I: CLINICAL TIME SERIES CLASSIFICATION

The first part of this thesis was dedicated towards the classification of clinical time series. First, we defined time series in a way that naturally extends to the intricacies encountered in real-world time series data (such as missingness, varying lengths of time series, incomplete observations, multi-dimensionality etc.). Then, we introduced different types of time series classification tasks. We distinguished between

1. whole-series classification, where a time series is classified as a whole (this was the case in Chapter 5),

2. window-based classification, where only a time window up until a certain time is used for classifying the time series (for example, a time window up until $n$ hours before sepsis onset, as described in Chapter 3), and finally

3. per-timepoint classification, where predictions are made at each time step of a time series, which was represented in Chapter 4.

We then gave an overview of existing ML approaches for time series classification. Even though we can organise these approaches into groups such as feature-based or distance-based, given the modularity and flexibility that recent deep learning frameworks provide, these boundaries have started to blur. Having introduced time series classification, we next introduced our application case of Part I, sepsis prediction.

Sepsis, a potentially fatal complication to infection, has been an age-old medical conundrum and currently represents a public health crisis [103]. Clinicians are challenged to diagnose sepsis in its early stages, when organ damage is still reversible, and where early interventions still can save lives [54]. However, due to sepsis being a heterogeneous syndrome, and since in its early stages, sepsis typically presents with unspecific signs and symptoms (e.g. confusion or fever), it remains notoriously hard to detect sepsis early. These circumstances make the early prediction of sepsis an interesting machine learning problem, that promises to be a task which can plausibly lead to clinical value. This is contrasted by efforts to predict endpoints that are easier to derive as well as easier to predict but may to some degree lack the perspective to provide clinical utility. To give an example, the prediction of diagnoses defined by billing codes is subject to several pitfalls: even though they are abundantly available in EHR datasets, they generally lack a temporal specification (as the billing code is assigned at the end or after the hospital stay)[1], are subject to interpretation biases and oblige monetary incentives [57, 155].

In Chapter 3 and 4, we presented two sepsis prediction studies. In the following, we recapitulate their findings, and put them into context.

---

[1] This could render an early prediction scenario futile.

### 7.1.1 UNCERTAINTY-AWARE RECOGNITION OF SEPSIS WITH GAUSSIAN PROCESS TEMPORAL CONVOLUTIONAL NETWORKS

In Chapter 3, we presented a single-centre study for the early prediction of sepsis, that for the purpose of this part of the dissertation served as a proof-of-concept study. We investigated whether vital and laboratory measurements at time windows before sepsis onset, compared to matched time windows in controls, are predictive of sepsis. Moreover, we addressed the data missingness problem by employing Gaussian process (GP) adapters, an uncertainty-aware neural network framework where as a first layer, a GP imputes the irregularly spaced and incompletely observed time series at evenly spaced time locations, and where subsequent layers[2] classify the imputed (or latent) time series. By introducing dilated causal convolutions to this framework, which exhibit a strong temporal inductive bias [7], we proposed MGP-TCN. In our experiments, we found that MGP-TCN outperforms MGP-RNN, the previous state-of-the-art method, in particular earlier than 4 hours before onset. We also observed that DTW-$k$NN, a classical distance-based classifier without employing deep learning achieved convincing results, even slightly outperforming MGP-TCN in terms of AUPRC at certain hours of the horizon analysis over 7 hours preceding sepsis onset.

In this chapter, we have encountered several core challenges: the creation of an annotated and reusable sepsis dataset, the implementation of hourly resolved sepsis labels that make an early prediction task possible, as well as methods to mitigate and leverage informative missingness. Nevertheless, we noted that the presented study has limitations. Our prediction setup, i.e., window-based classification using data up until $n$ hours before (matched) sepsis onset, reveals whether there are pre-onset signals predictive of sepsis. However, this setting is different from a prospective evaluation, where we do not know in advance when a sepsis onset will occur, and where models are tasked to continuously monitor patients in order to raise early alarms. Second, the setting of this study allowed for the inclusion of methods such as GP adapters or DTW-$k$NN that may be efficiently trained for time series classification (whole series or window-based), but which do not scale beneficially when reformulating the prediction problem into an online prediction task. Notably, even though MGP-TCN is a sequence-to-sequence model that during training could output predictions at each time location of the input data, these predictions would not be *causal* in that the GP would leverage data from the future. As elaborated in Section 3.6, we expect that it could be an exciting route for future work to develop and employ online variants of the methods investigated in this chapter. We envision that the scalability of the DTW-$k$NN approach could be improved if during training a disentangled subset of patients (or even latent time series templates) was

---

[2] They can be interpreted to represent a downstream classifier.

learned that are used for the nearest neighbour search step upon prediction which could achieve a runtime complexity constant in the number of time series (i.e., patients) in the training set. The scalability of GP adapters could be further improved by exploring alternative GP formulations beyond standard approximation schemes, for instance with state-space GPs that could lead to a runtime that is linear in the number of training points [188].

### 7.1.2 PREDICTING SEPSIS IN MULTI-SITE, MULTI-NATIONAL INTENSIVE CARE COHORTS USING DEEP LEARNING

In Chapter 4, we conducted a multi-centre study for the prediction of sepsis. This involved the harmonisation and annotation of five ICU datasets that together represent the first international multi-centre cohort for sepsis prediction. Among the previously listed challenges, this chapter was foremost focused on validity and datasets. Specifically, we were wondering whether we can train ML models for the early prediction of sepsis that generalise to new, previously unseen sites. This was motivated by our systematic review on sepsis prediction [143], where we found that most sepsis prediction studies did not (and plausibly could not) perform an external validation. In order to conduct this validation, we harmonised ICU data from five databases, resulting in the to-date largest, international dataset for sepsis prediction. In terms of prediction methods, we then devised a deep self-attention model and included several baseline ML models, as well as several clinical baseline scores. In an extensive internal and external validation, we found that the attention model indeed can generalise to unseen sites when leveraging a federated learning setting, where models trained on different sites are pooled upon external testing on a new site. We further highlighted that given certain non-reducible heterogeneities between the datasets (alternative suspected infection implementation, differences in the cohort composition, etc.), it is not surprising that training and testing on pairs of datasets exhibited more moderate performance than the federated approach. Moreover, we observed that pooling on the model level even led to superior performance than pooling on the dataset level, which required the costly retraining of models for each combination of datasets to be pooled.

In this project, we provided a platform for performing external validations on a multi-centric ICU cohort featuring data from three countries: the US, the Netherlands, and Switzerland. It is the hope of the authors of this study, that this platform will facilitate further validation studies, which are urgently needed as even deployed models have been shown to be insufficiently validated [211]. Furthermore, the findings of this study are well-aligned with recent studies that employ federated learning to leverage multi-centric data in a differentially private manner [46, 173]. Albeit we have created and analysed a multi-centric cohort for sepsis prediction in this chapter, we still have only scratched the surface. For instance, given the employed online prediction task which was designed to closely reflect a real-time deployment scenario, the next step will be to *prospectively* validate our models to check whether early alarms for sepsis indeed lead to a benefit for the hospitalised patients. Previously, a small randomised controlled trial indeed suggested a beneficial effect on mortality and length of stay for the InSight model, a closed-source proprietary risk model [183]. Furthermore, a relevant limitation of the presented multi-centre dataset is the foremost Caucasian cohort, collected in western countries. Developing sepsis prediction models on more inclusive cohorts and validating them also on data distributions from non-western countries will considerably boost the global relevance, generalisability, and applicability of such models. Another interesting line of follow-up research will be to more closely consider the domain adaptation problem, and to characterise transfer failure modes, i.e. to identify situations where model transfer does not work, in order to develop dedicated mitigation strategies.

Regarding the experimental setup, one interesting area for future work will be the specification of the prediction target. Morrill et al. [145] have proposed to replace binary prediction targets with a continuous target (based on a domain-specific utility score) transforming the sepsis prediction problem into a regression problem. While such continuous targets may account for more granularity (e.g., how useful it would to raise an alarm at which time), the underlying utility functions are essentially hand-crafted and are challenging to define in an objective manner [166]. As a further obstacle, we observed that this existing utility score for sepsis does not generalise to new datasets with different dataset statistics (prevalence of sepsis, length of stay, etc.) [166]. It is a vision of the author of this thesis, to leverage the domain knowledge of clinical experts in a statistically sound way to create novel utility scores that allow for more fine-graded supervision during training, while explicitly depending on dataset statistics, such that it can be easily adapted to new data distributions. Furthermore, it will be interesting to consider strategies to improve calibration already during training. For instance, overconfident alarms could be counteracted with label smoothing strategies [146].

As a notable take-away from this study, we conducted a successful external validation *despite* label shifts between the datasets. This suggests that our models exhibit a certain robustness even under moderate changes in the label implementation. Counterfactually, this finding could make it easier to generalise to new cohorts from countries not included in the current dataset, where a certain label and dataset shift is to be expected. Given that our federated learning strategy showed convincing results—even compared to the pooling of the underlying data—we envision that clinical prediction models could be scaled *massively* via the distribution of i) a standardised protocol for collecting and preprocessing the data, ii) code for the development of local models, and iii) pretrained models for local evaluations. As opposed to centralised multi-centre studies (as the one conducted in this thesis), in such a decentralised scenario it would be drastically easier to guarantee the safety of patient data as it never had to leave the walls of the respective source hospital. While a decentralised training and sharing of models is rather straightforward, decentralised *preprocessing* and data harmonisation may turn out to be more challenging, as preprocessing and data cleaning strategies are frequently devised on the fly to account for dataset-specific peculiarities and artefacts. Nevertheless, we hope that our finding may serve as further piece of evidence to motivate federated learning in order to better protect sensitive patient data.

## 7.2 PART II: TEMPORAL AND TOPOLOGICAL REPRESENTATION LEARNING

In Part I we took an "application-centred" view, focussing on early predictions in the clinical context. In contrast, Part II of this dissertation is taking a "model-centred" view. We explored data representations that deep neural networks learn by considering two perspectives. First, in Chapter 5 we investigated representations that are learned on time series. In Chapter 6, we developed a more general framework for the learning of low-dimensional representations that aim to preserve the structure of the data space. These two chapters are complementary, not only in that the first one attends to time series, while the second one refers to any data space equipped with a distance measure. Also, the first chapter considers representations in an instrumental way, i.e., to optimise a downstream classification task, while in the second chapter of Part II, we intrinsically want to learn faithful representations that reflect and preserve the data space.

### 7.2.1 PATH SIGNATURES FOR TIME SERIES REPRESENTATION LEARNING

Our Chapter 5 was dedicated to path signatures, a rich and theoretically well-studied transform rooted in rough path theory. We elaborated on several interesting properties of the signature, such as i) *uniqueness*, i.e., any path is fully determined by its signature, or ii) the signature being a *universal non-linearity*, in that every function of a path can be arbitrarily well approximated by a linear map on the signature. Recent studies employing the signature observed beneficial effects in terms of predictive performance [19, 145]. However, despite a rich theory and successful applications, in this chapter we found that the signature is plagued by an issue that has been neglected so far. Specifically, the signature acts on continuous paths, but in practice is only provided discrete time series observations such that continuous paths are constructed *implicitly*. We proposed to make this step explicit by formulating this step as a *path imputation*, and considered several path imputation strategies. Experimentally, we classified irregularly spaced time series using signature models and found that signature models are indeed impacted by the choice of path imputation, in particular more shallow models. By proposing a novel GP adapter variant, GP-PoM, which allows for the propagation of uncertainty at each prediction step, we improved the robustness of signature models when dealing with irregular time series, thereby proposing a mitigation strategy for the identified problem.

In this chapter, we encountered the challenge of data missingness and showed that when dealing with it implicit choices can have a drastic effect on performance, in particular when imputing a continuous path of data. While the task of converting discretely observed data into a continuous path in data space may seem particularly relevant when dealing with signatures, one could argue that a related phenomenon is also occurring in the context of other modelling architectures, such as convolutional or recurrent neural networks. For instance, even though convolutions are frequently considered as discrete sums, in essence, they merely represent a numerical quadrature of the continuous integral cross-correlation between a path in data space with some learnable filter. There is initial work to explore continuous formulations to better model irregular data [55], but meanwhile, with regular convolutions we are implicitly treating data observations as a continuous path in data space. Similar connections between recurrent architectures and their continuous analogues can be made [110]. In conclusion, for signature-models as well as for non-signature models, the implicit usage of data as a continuous path evolving through data space seems to omnipresent whilst also being swept under the rug for convenience. While in this chapter, we have focussed on signature models and the time series domain, we nevertheless have demonstrated that this reoccurring pattern deserves further attention, beyond the analysis of temporal data.

For future work, it will be an exciting to route to apply signatures to paths in low-dimensional latent spaces, to paths on time-varying graphs, or to paths that arise in computer vision (changing poses) or computer graphics (paths of rendered objects). Initial works relating the signature with persistent homology suggests that there is a rich interplay to be further investigated [39], for instance to recover a fixed-dimensional representations of paths in topological feature spaces. As a further area of development, signatures may be used to construct time series kernels [113]. Given Chen's property, i.e., the fact that the signature of two concatenated paths (or time series subsequences) can be efficiently computed using the signatures of the individual paths, we envision that kernels acting on time series subsequences could be efficiently augmented using the signature [15].

### 7.2.2 TOPOLOGICAL REPRESENTATION LEARNING

In Chapter 6, we considered learned representations through the lens of topological data analysis. We noted that representation learning may not only be directed towards optimising a downstream task, but that it can also be of interest to learn embeddings that reflect the structure of the input space—be it to learn the structure of the data manifold, or to reduce the dimensionality in high-dimensional datasets that are challenging to visualise. In this work, we proposed a novel differentiable loss term that incentivises autoencoders to learn low-dimensional embeddings that preserve topological features of the data space. By imposing the (weak) assumption that persistence diagrams have unique entries (i.e., we observe unique distances in the Vietoris–Rips complex (VR) filtration), we showed that this loss term is indeed differentiable and can be integrated in a neural network architecture that is trained using backpropagation. Our experiments showed that this method, topological autoencoder (TopoAE), was the only method that could preserve complex nesting behaviours of sphere manifolds. On real-world image datasets, we observed that TopoAE led to favourable embeddings that best preserved density estimates of the data space.

The presented method is defined in a generic way; we merely require data objects and a distance function. Therefore, for future work it will be interesting to apply this method to different types of data such as time series, biological sequencing data, or graph-structured data. While the topological loss of TopoAE considers the structure that emerges in point clouds of data points, an interesting alternative view point will be to learn topological features of individual data points. For instance, the author of thesis has contributed to a recent study that proposed a topology-aware graph neural network layer that increases the expressivity of conventional graph neural networks [92]. We further envision that topological representation learning will offer exciting new perspectives for machine learning on time series, be it via visibility graphs, level set topologies, point cloud representations of time series (for instance, via delay embeddings), or by directly integrating topological features into time series models.

## 7.3 OUTLOOK

In this dissertation, we first investigated the classification of clinical time series using machine learning. To this end, we considered the prediction endpoint, sepsis, the early identification of which remains challenging and promises clinical value. Across the first part of this thesis, we observed and addressed key challenges that the sepsis prediction literature has been facing. Most prominently, we presented a harmonised multi-centric dataset that allowed us to conduct external validations across countries which the sepsis prediction literature has been systematically neglecting due to the lack of access to annotated validation data.

Moreover, in this thesis we have encountered data missingness problems, and have proposed mitigation strategies by leveraging Gaussian processes that are trained by a downstream task. In the second part of the dissertation, we further investigated how models actually interpret and represent input data, first with the example of irregularly spaced time series. For that, we learned time series representations using path signatures, a powerful framework for encoding paths of data. Furthermore, we observed that the application of this transform carries relevant but implicit decisions about how the raw and irregularly observed input data is interpreted. Again leveraging the uncertainty awareness of Gaussian processes, we developed a strategy to make models employing the signature, which thereby work on paths, more robust. As a final outlook of this thesis, we aimed to learn representations that not only account for irregularities in the input data, but that actually *preserves* relevant structures in the input data. For this, we leveraged tools from topological data analysis and devised a novel autoencoder variant that naturally preserves structures (in the sense of topological features) of high-dimensional data spaces in latent encodings, thereby revealing complex manifold structures that were hard to access with existing methods.

In the following, we briefly outline future directions which were motivated by the content and findings of this dissertation. With increasing dataset sizes, it becomes harder to manually examine a significant portion of the data. Thus, biases in the data may be further amplified when training black-box models on such datasets. The typical clinical ML model is unable to distinguish causation from correlation, and will exploit any correlation it can find to optimise its training objective. However, spurious correlations and confounding can lead to useless or even harmful models, which has become a foundational challenge across disciplines that employ ML. These challenges may be partially addressed using causal modelling strategies. However, in particular for clinical time series, we argue that instead of *inferring* causal effects from observational data (for instance via Granger causality), we hypothesise that we need to *enforce* causality using domain knowledge. To give a practical example, a model may have monitored hundreds of years worth of hospital stays, but is still oblivious to the fact that vasopressors increase mean arterial blood pressure (MAP). As vasopressors are typically administered to hemodynamically stabilise patients, it is possible that when MAP would drop, an increased vasopressor dosage keeps MAP constant, i.e., these two variables may not even appear correlated. Inferring this relationship from observational data alone—which additionally is swamped with noise and interactions—is nigh impossible. However, we argue that this is not necessary. By including domain experts in the model development process (i.e., "human in the loop"), we envision models that *know* about established relationships between relevant variables before having observed a single training data point. This may be achieved by means of encoding prior causal knowledge in a directed knowledge graph, that a prediction model may consult during the prediction step.

Regarding the application domain sepsis, we see two main focus areas. First, given the complexity of the Sepsis-3 definition, and the dependence on actions by the clinician (ordering of cultures, administration of antibiotics), a more data-driven definition of the onset of sepsis would be desirable. This may be achieved by formulating the prediction task as an unsupervised outlier detection problem, or by performing a weakly supervised classification task, where only patient-level class labels are available during training and the goal is to correctly classify patients as early in their stay as possible, thereby elucidating early signs of sepsis in a data-driven manner. Second, given its success in various other application domains, we expect that self-supervised learning strategies could improve the learned time series representations [97], for instance by means of learning to classify basic patient states (e.g. fever) as auxiliary tasks, which could lead to beneficial downstream predictive performance.

For representation learning, a central focus for the future work will be to translate our findings into applications. For instance, the TopoAE could reveal insightful structures of high-dimensional biological sequencing data. However, we expect that depending on the exact application case, the loss term or the employed data space distance may need to be modified. On a larger scale, the author of this thesis is eager to identify further representation learning scenarios where topological features are relevant while currently being neglected due to their non-trivial computation. Having employed different types of non-parametric layers inside deep neural networks in this thesis (Gaussian process layers, signature layers, and topological layers), it will be interesting to characterise in which cases they lead to beneficial representations, and in which scenarios they are actually detrimental.

Faithful visualisations of high-dimensional data have become an essential means to interpret data in various disciplines of the life sciences. Therefore, we foresee that topology-aware representations will lead to more interpretable embeddings and robust analyses, where global structures in the data spaces are properly accounted for. In contrast, currently established and widely used methods such as t-SNE or UMAP preserve foremost local neighbourhoods and are sensitive to hyperparameter choices, which has sparked an on-going controversy about the validity of biological analyses based on these methods [28].

Motivated by increasingly large, multi-modal and high-dimensional biological datasets, several dimensionality reduction methods have been introduced over the last two years. To name a few examples, this includes PHATE [137], a heat-diffusion based approach that allows for embeddings that recover tree-like hierarchies in high-dimensional single-cell data. Another work proposed ivis, a deep learning-based dimensionality reduction method that employs a contrastive triplet loss for learning structure-preserving embeddings of single-cell data [192]. A next study leveraged Poincaré maps, a tool from hyperbolic geometry, to account for hierarchies in biological data [114]. Most recently, Zhou et al. [221] proposed GraphDR, a quasi-linear method that preserves neighbourhood relations in data represented as graphs. Amidst a growing list of new dimensionality reduction methods, it is currently not clear which perspective or angle is useful or detrimental for which application in biomedical datasets. Aggravatingly, these new methods have been compared against established flagship methods such as t-SNE and UMAP, but not against each other. Therefore, we envision that a careful comparison of this wave of new dimensionality reduction techniques will be an exciting and necessary effort for future work. Clarifying which framework (topology, diffusion processes, contrastive learning, hyperbolic geometry, etc.) is indeed beneficial to which application niche will greatly illuminate and advance the analysis of high-dimensional datasets of biomedical systems.

# SUPPLEMENTARY MATERIALS

## A.1 PREDICTING SEPSIS IN MULTI-SITE, MULTI-NATIONAL INTENSIVE CARE COHORTS USING DEEP LEARNING

### A.1.1 RESULTS ON THE EMORY DATASET

In Figure A.18, we display the internal validation results on the Emory dataset. Furthermore, we also applied pair-wise external validations by retraining all models on the smaller set of 35 variables in order to transfer across datasets (from Emory or to Emory). These results are shown as part of Figures A.8 to A.17.

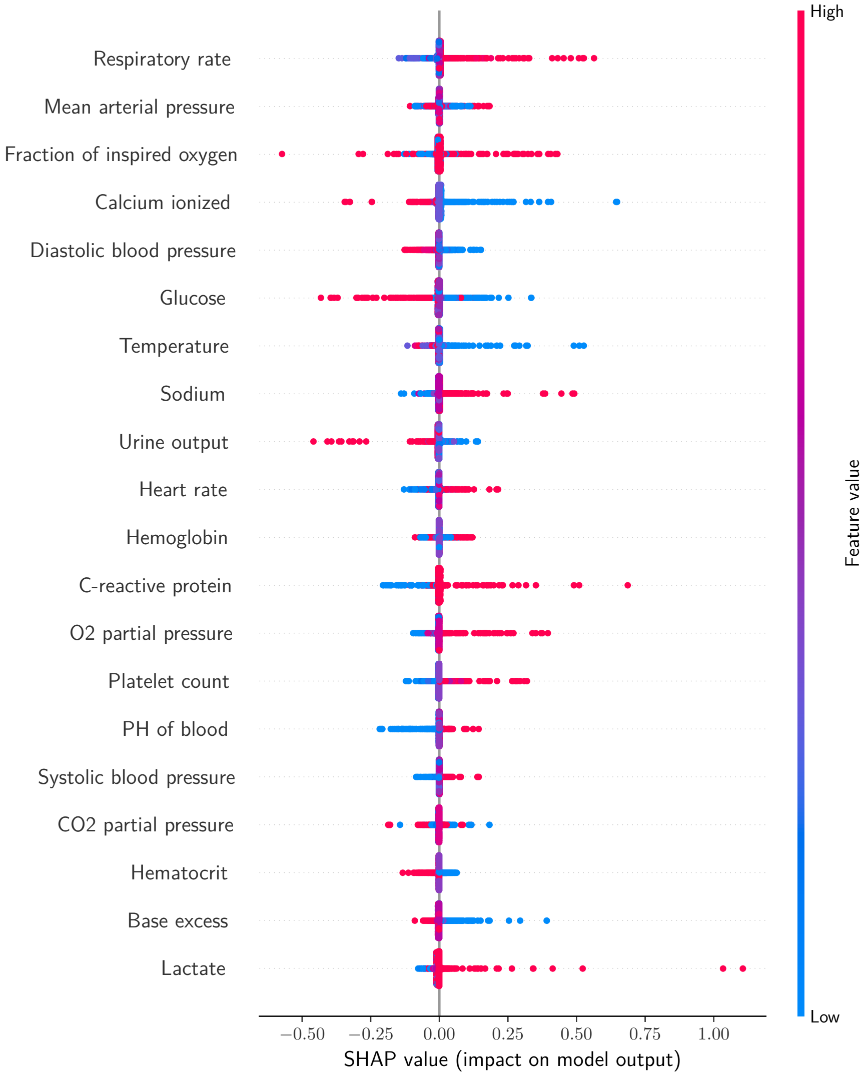


Figure A.1: Shapley value distributions on the AUMC dataset. The 20 variables with the largest mean absolute Shapley values are shown.

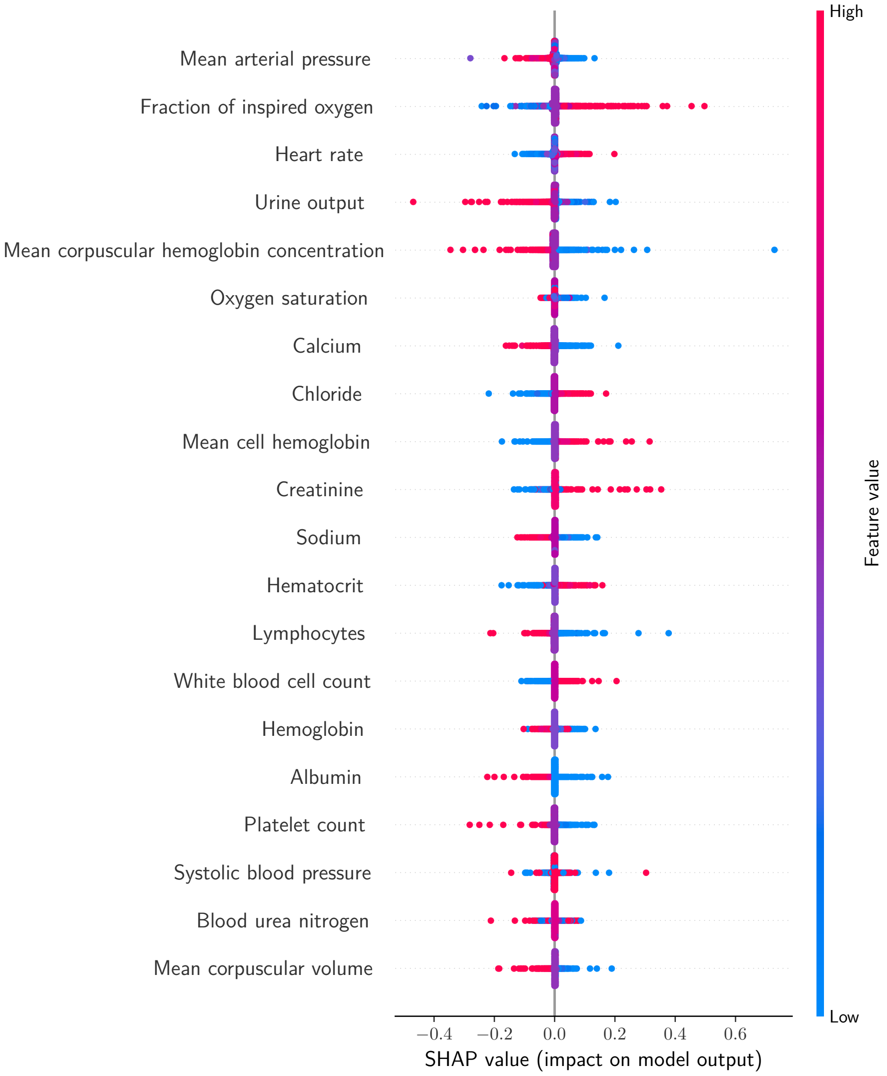


Figure A.2: Shapley value distributions on the eICU dataset. The 20 variables with the largest mean absolute Shapley values are shown.

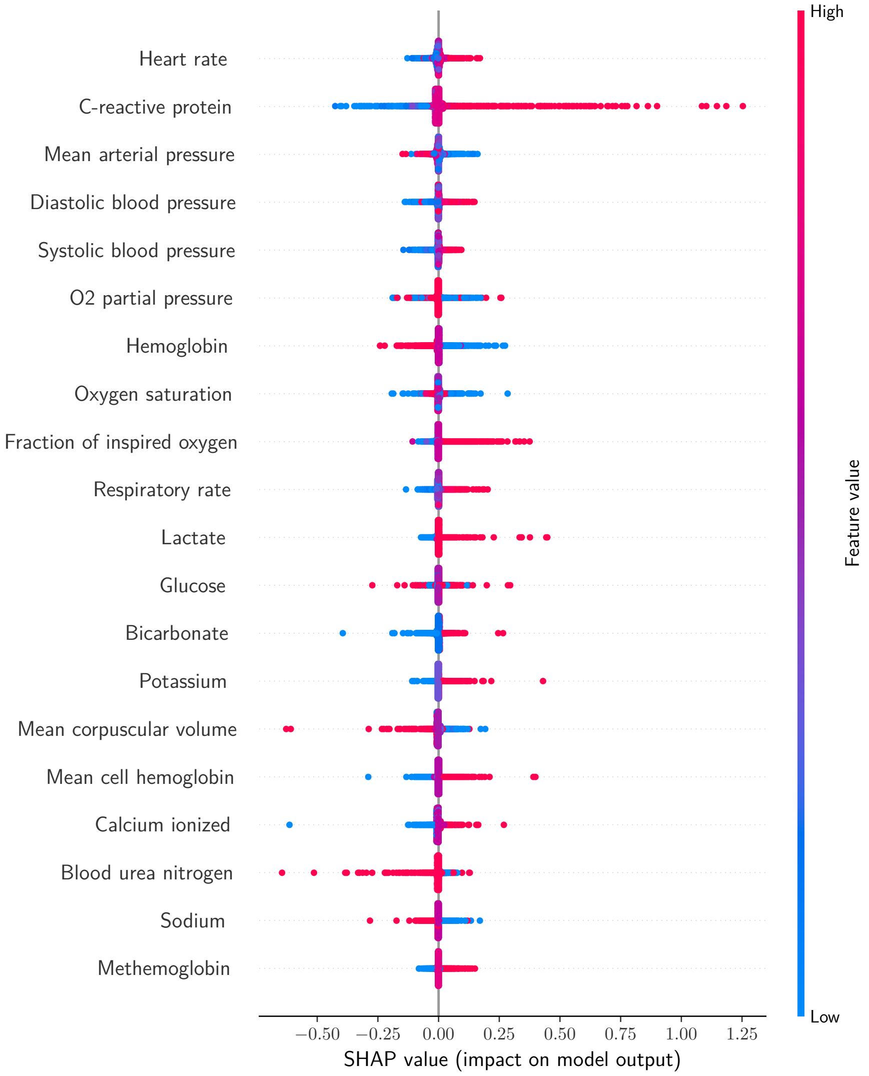


Figure A.3: Shapley value distributions on the HiRID dataset. The 20 variables with the largest mean absolute Shapley values are shown.

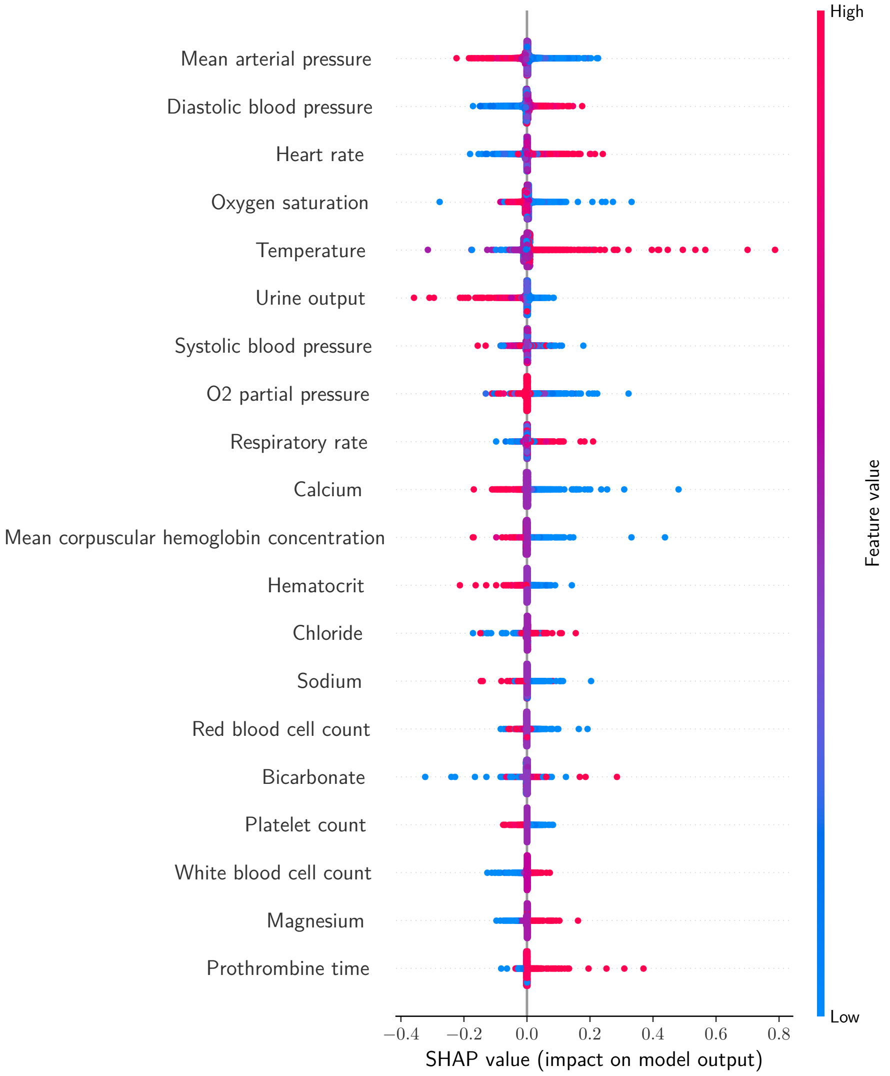


Figure A.4: Shapley value distributions on the MIMIC-III dataset. The 20 variables with the largest mean absolute Shapley values are shown.

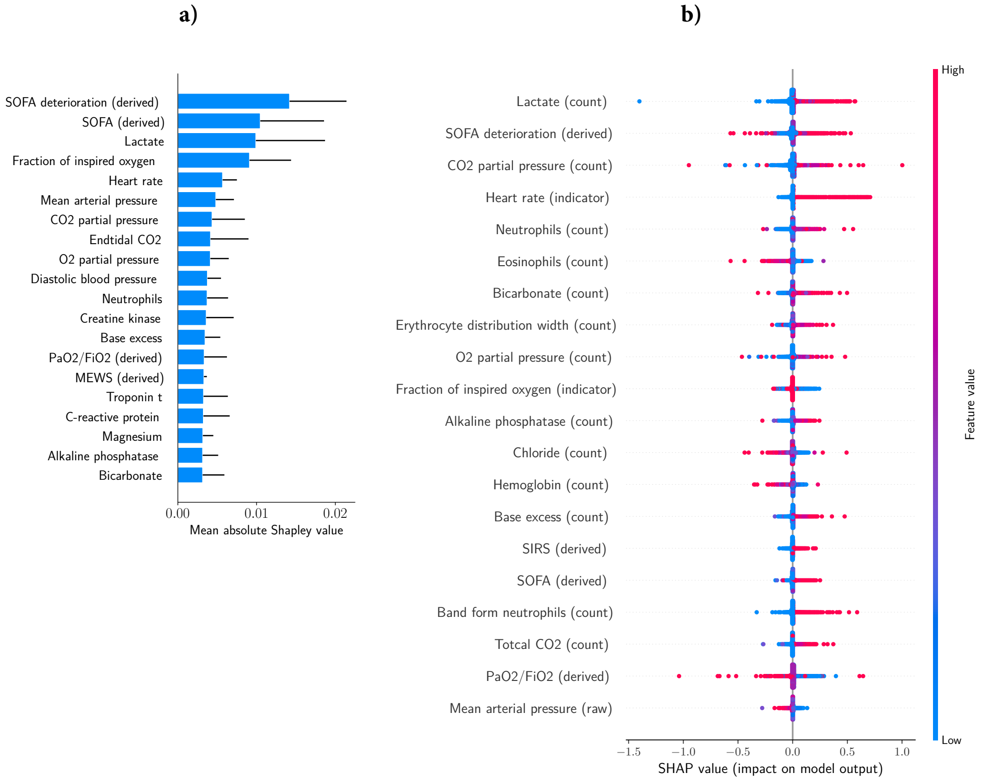


Figure A.5: Shapley analysis that includes all feature types that were used in the attention model, raw measurements, measurement counts, missingness indicators, and derived features. While Panel a) illustrates the 20 features with the largest mean absolute Shapley values averaged across all core datasets, Panel b) exemplifies the Shapely value distributions on the eICU dataset.

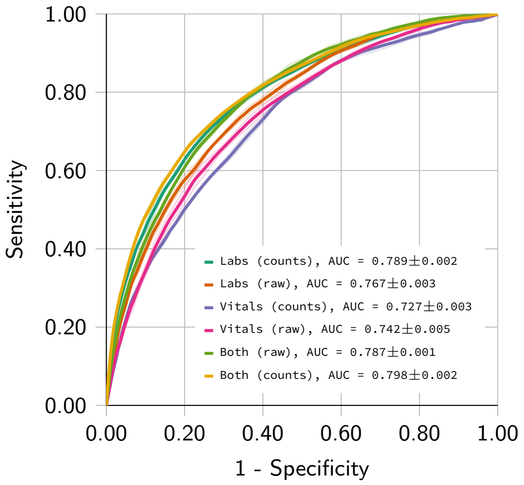


Figure A.6: Ablation of features. Given that we observed that count features frequently appeared as top ranking variables in Figure A.5, we investigate how well an attention model performs when only trained on either raw observations or on measurement counts. We further subdivide into only using vital signs, laboratory measurements, or both. This analysis was performed on MIMIC-III (both training and testing). While we found no striking difference in performance, we observe the trend that for irregularly measured lab values counts are indeed informative, while this was less the case for frequently monitored vital signs.

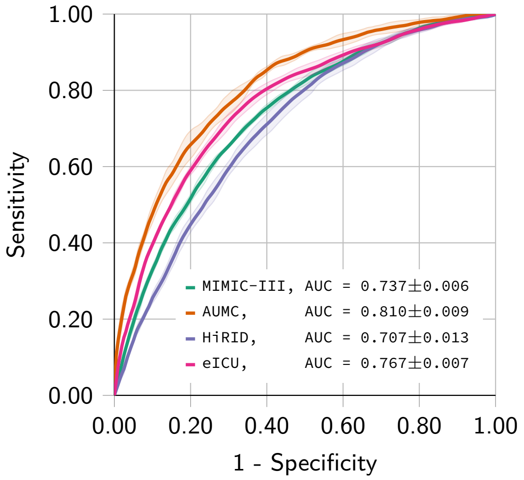
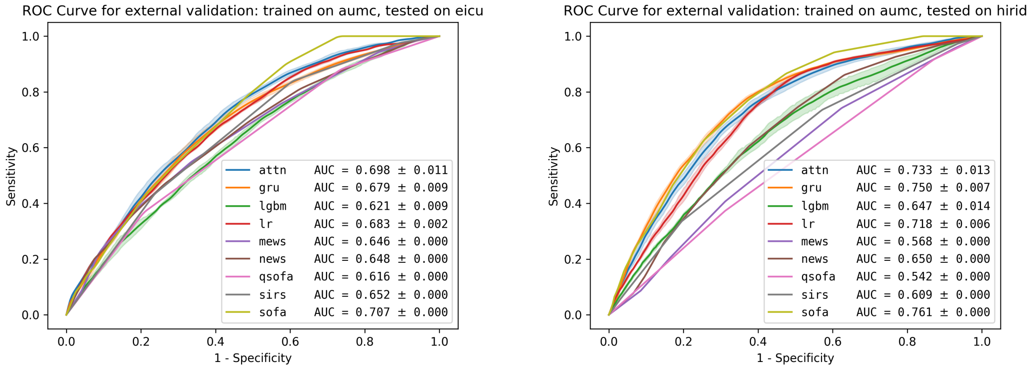


Figure A.7: ROC plots for the auxiliary analysis, where the attention model was trained by pooling the data from the different datasets already during *training*, in contrast to pooling only the predictions of models that were trained on separate datasets. Here, for a given dataset the remaining datasets were pooled for training.

**a)** eICU **b)** HiRID



**c)** MIMIC-III **d)** Emory

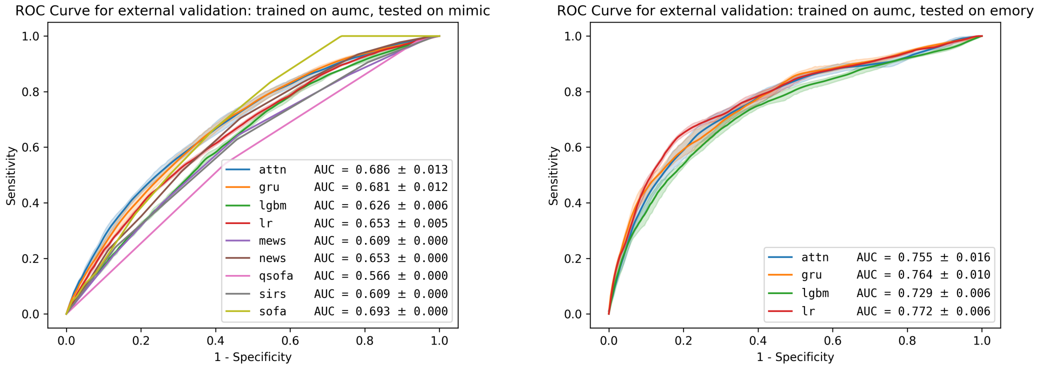
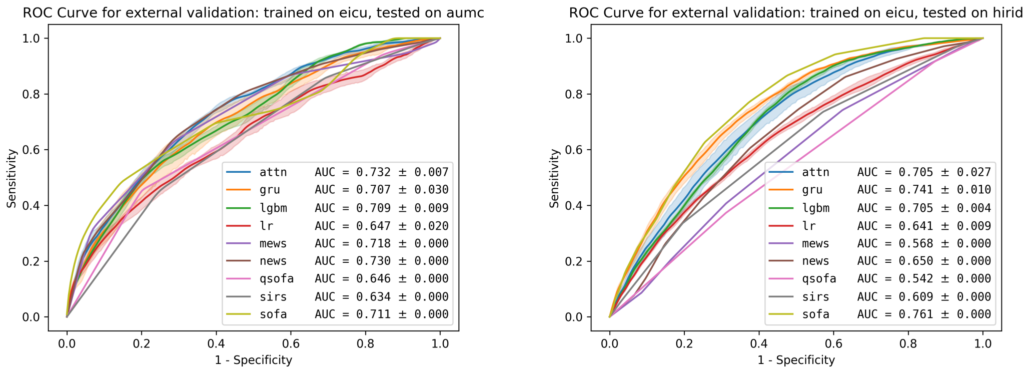


Figure A.8: ROC curves for the pair-wise external validations. All displayed models were trained on the AUMC dataset and applied to one of the remaining datasets, as indicated in the figure heading.

**a)** AUMC **b)** HiRID



**c)** MIMIC-III **d)** Emory

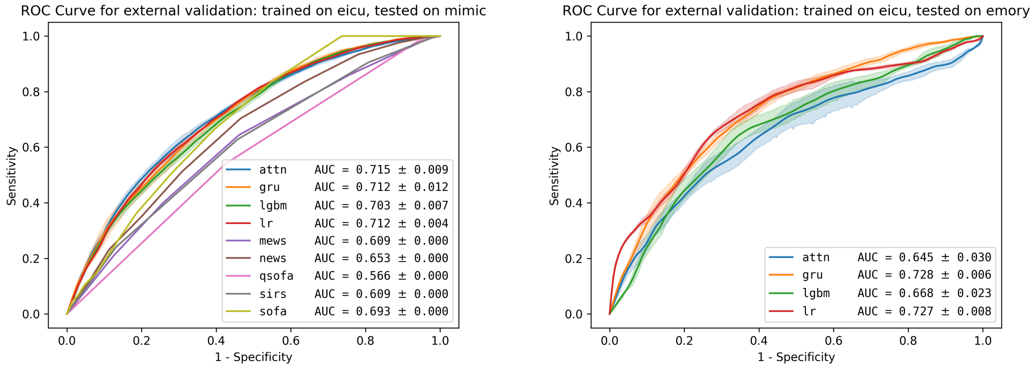
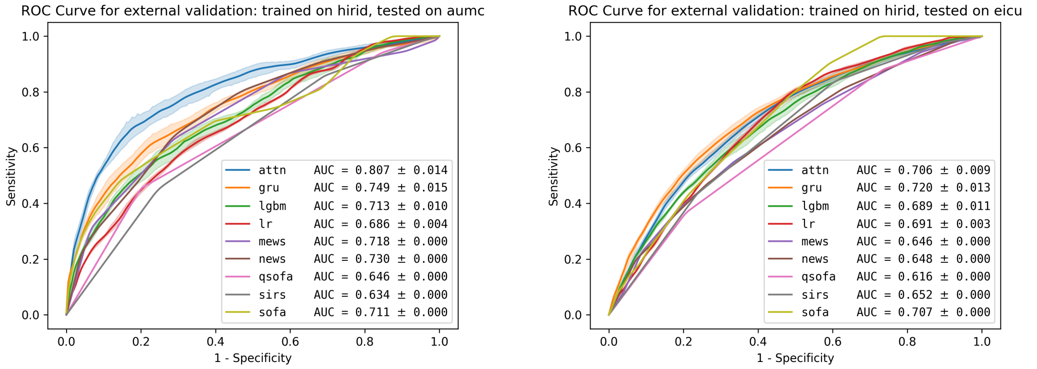


Figure A.9: ROC curves for the pair-wise external validations. All displayed models were trained on the eICU dataset and applied to one of the remaining datasets, as indicated in the figure heading.



**a)** AUMC

**b)** eICU

**c)** MIMIC-III

**d)** Emory

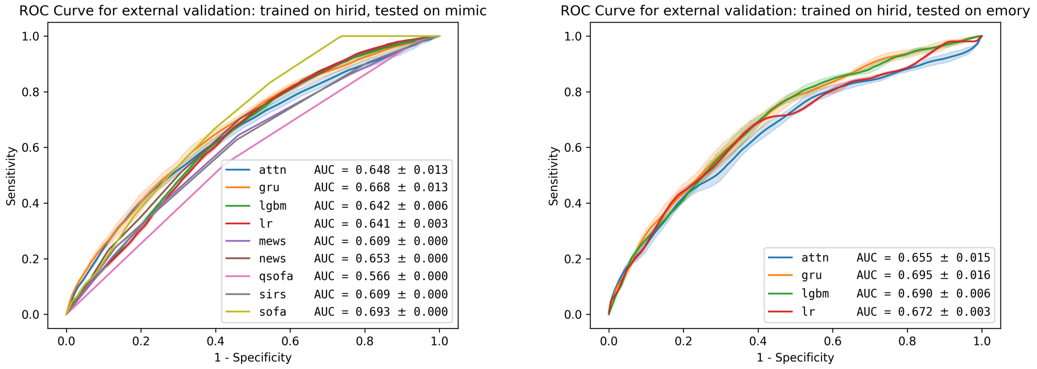
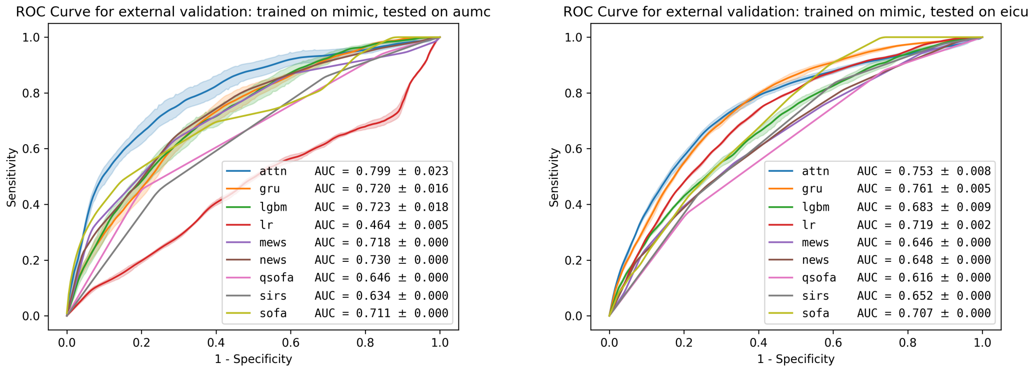


Figure A.10: ROC curves for the pair-wise external validations. All displayed models were trained on the HiRID dataset and applied to one of the remaining datasets, as indicated in the figure heading.



**a)** AUMC **b)** eICU

**c)** HiRID **d)** Emory

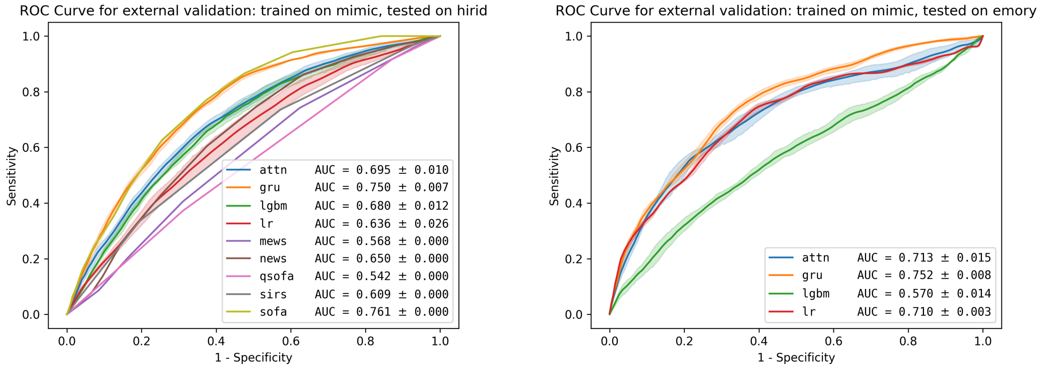
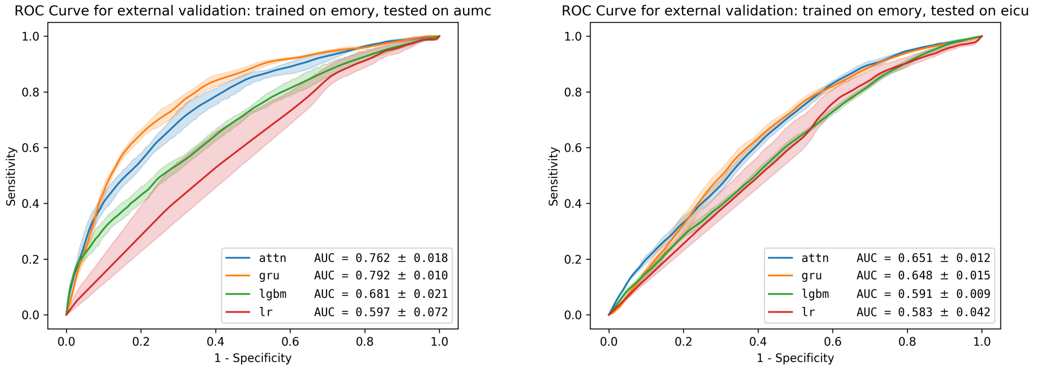


Figure A.11: ROC curves for the pair-wise external validations. All displayed models were trained on the MIMIC-III dataset and applied to one of the remaining datasets, as indicated in the figure heading.

**a)** AUMC

**b)** eICU



**c)** HiRID

**d)** MIMIC-III

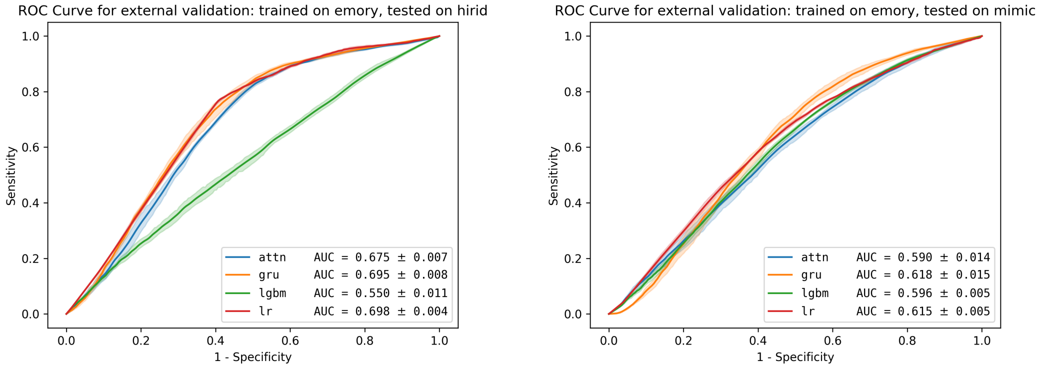


Figure A.12: ROC curves for the pair-wise external validations. All displayed models were trained on the Emory dataset and applied to one of the remaining datasets, as indicated in the figure heading.

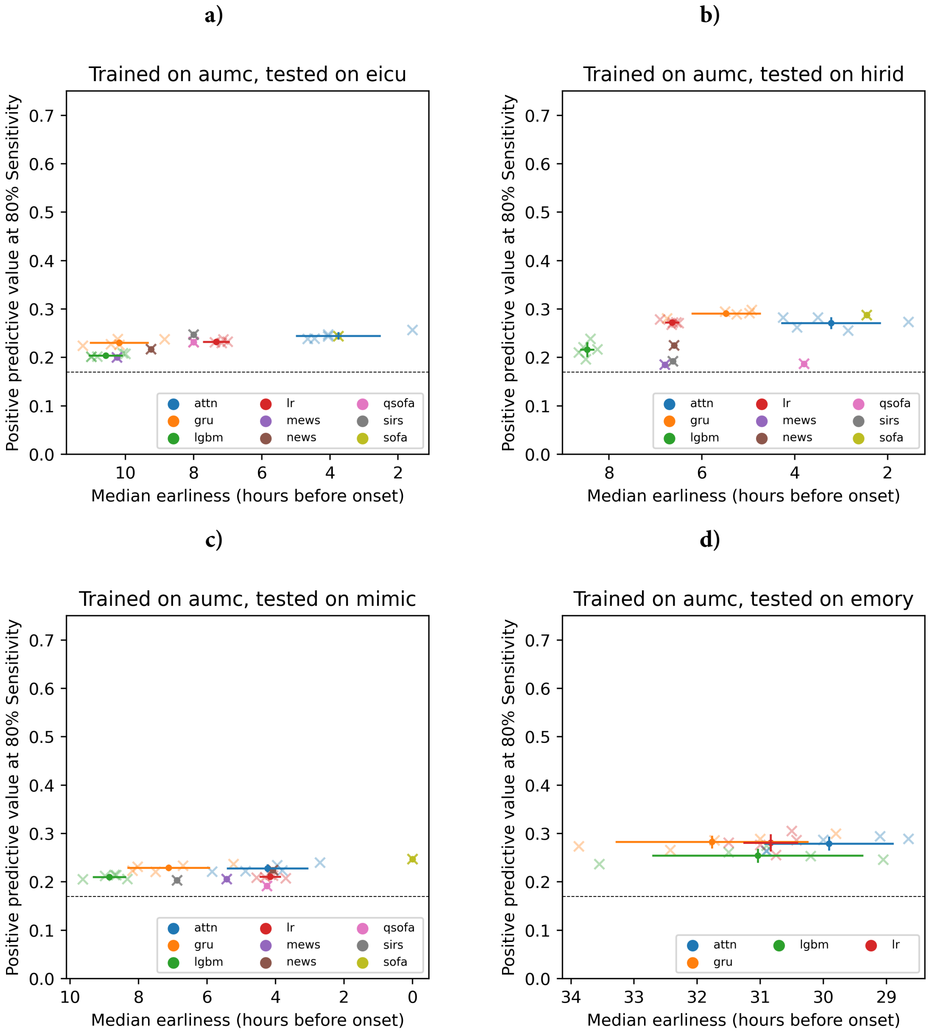


Figure A.13: Scatter plots for the pair-wise external validations. All displayed models were trained on AUMC dataset and applied to one of the remaining databases, as indicated in the figure heading.

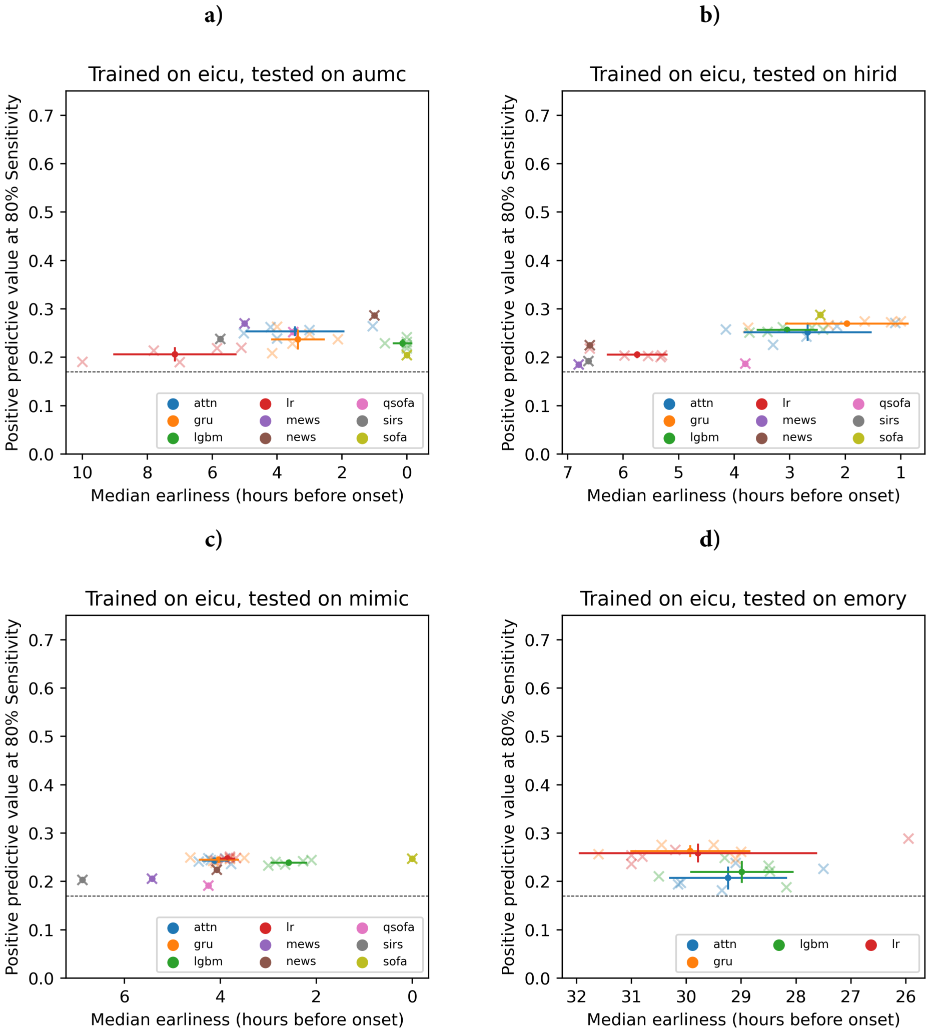


Figure A.14: Scatter plots for the pair-wise external validations. All displayed models were trained on eICU dataset and applied to one of the remaining databases, as indicated in the figure heading.

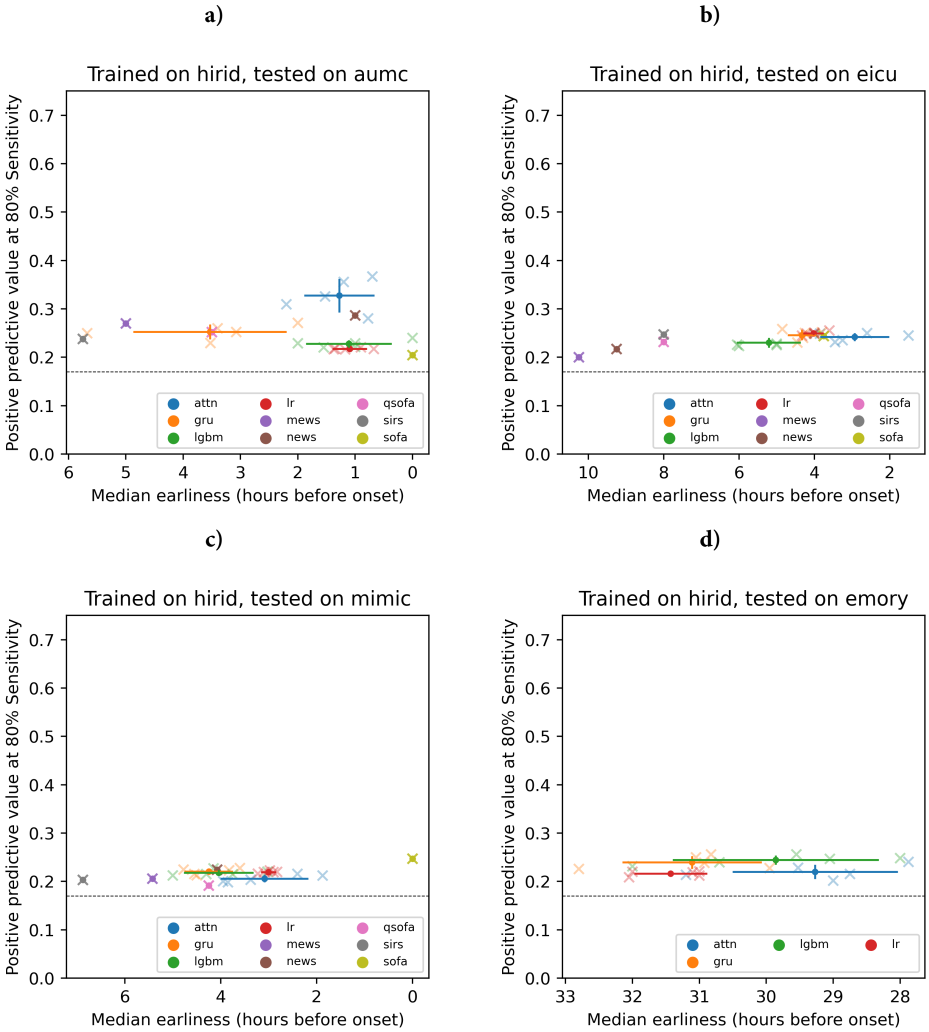
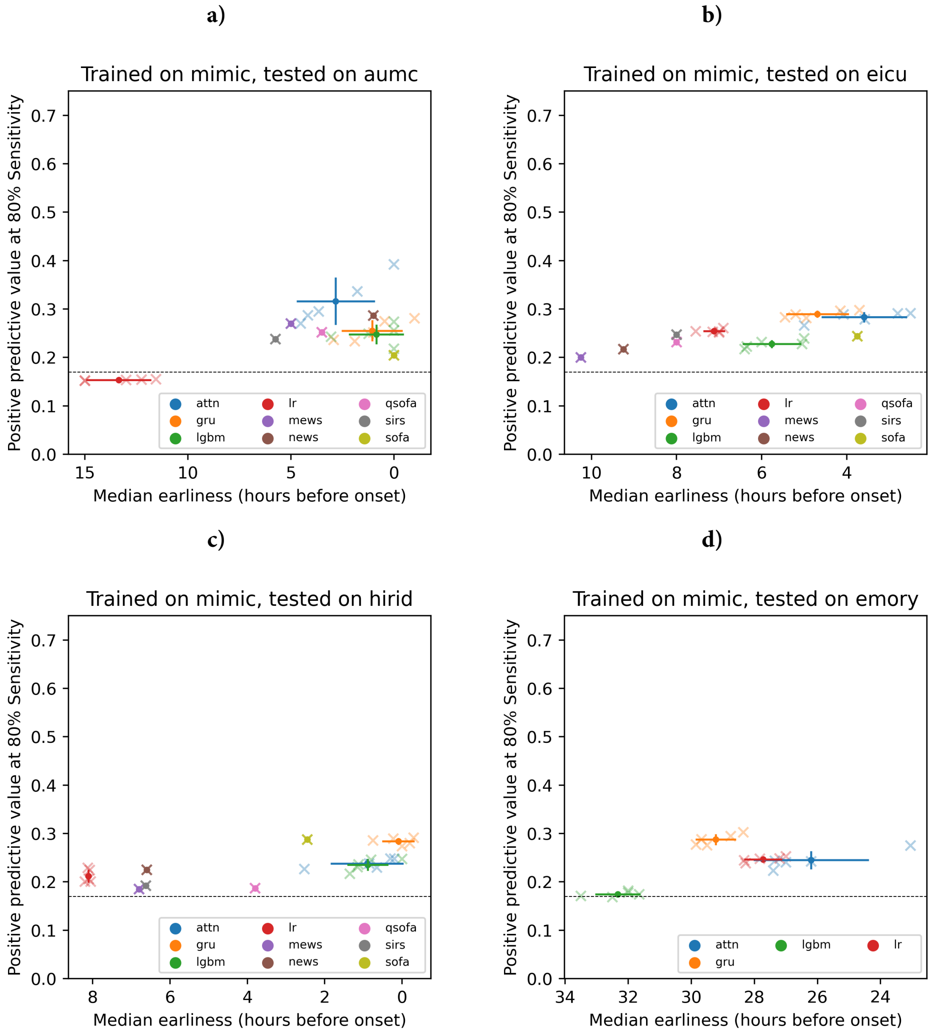


Figure A.15: Scatter plots for the pair-wise external validations. All displayed models were trained on HiRID dataset and applied to one of the remaining databases, as indicated in the figure heading.



a) b) c) d)

Figure A.16: Scatter plots for the pair-wise external validations. All displayed models were trained on MIMIC-III dataset and applied to one of the remaining databases, as indicated in the figure heading.

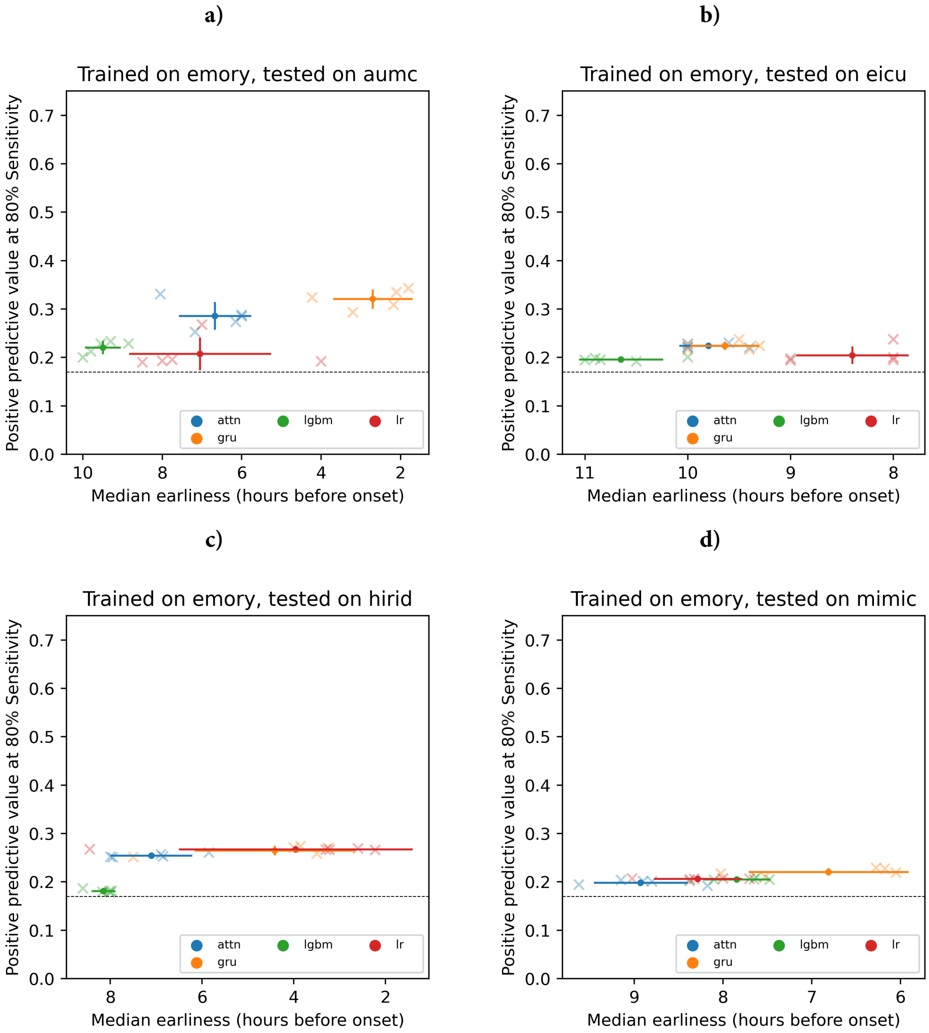


Figure A.17: Scatter plots for the pair-wise external validations. All displayed models were trained on Emory dataset and applied to one of the remaining databases, as indicated in the figure heading.

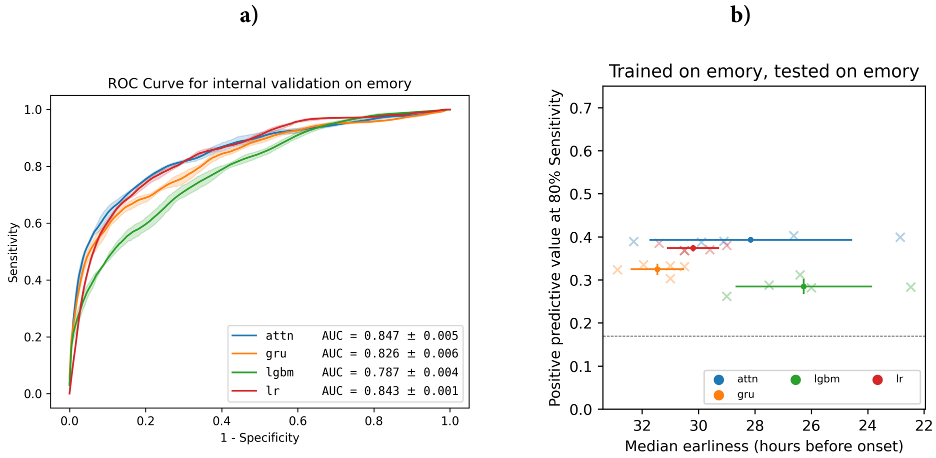


Figure A.18: Internal validation on the Emory dataset. Panel **a)** displays ROC curves, Panel **b)** shows the alarm earliness plot. This dataset was published in a preprocessed and annotated stage, reporting only a smaller variable set. Therefore, due to missing information the clinical baseline scores were not extracted on this dataset.

## A.2 PATH SIGNATURES FOR TIME SERIES REPRESENTATION LEARNING

Tables A.1 and A.2 show additional results for the CHARACTERTRAJECTORIES and PENDIGITS datasets under *random* subsampling of the time series.

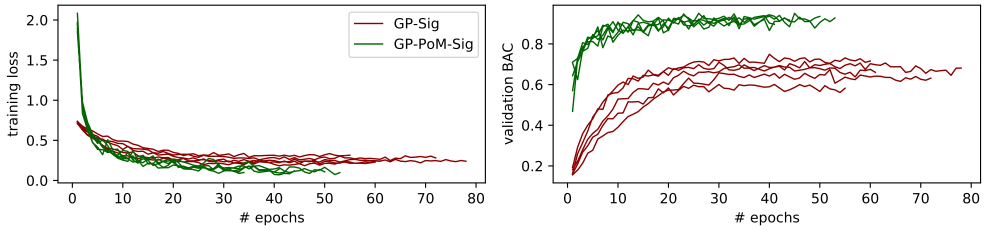


Figure A.19: Comparison of the training of the GP-PoM and the standard GP adapter, illustrated for the CHARACTERTRAJECTORIES dataset and the SIG model.

Table A.1: CharacterTrajectories, under random subsampling, $p = 50\%$

| Imputation | Model | w-AUROC | BAC | Accuracy |
|---|---|---|---|---|
| GP-PoM | DeepSig | $99.698 \pm 0.393$ | $94.011 \pm 5.037$ | $93.635 \pm 5.335$ |
| | RNN | $99.970 \pm 0.011$ | $98.011 \pm 0.512$ | $98.106 \pm 0.508$ |
| | RNNSig | $99.787 \pm 0.074$ | $93.308 \pm 0.960$ | $93.844 \pm 0.903$ |
| | Sig | $99.578 \pm 0.031$ | $89.570 \pm 0.938$ | $89.930 \pm 0.914$ |
| GP | DeepSig | $98.994 \pm 1.088$ | $90.821 \pm 2.361$ | $90.471 \pm 2.347$ |
| | RNN | $99.909 \pm 0.032$ | $96.276 \pm 0.691$ | $96.492 \pm 0.715$ |
| | RNNSig | $99.400 \pm 0.094$ | $87.587 \pm 2.054$ | $88.141 \pm 1.959$ |
| | Sig | $94.862 \pm 1.779$ | $61.280 \pm 6.440$ | $62.446 \pm 6.493$ |
| causal | DeepSig | $99.963 \pm 0.023$ | $97.774 \pm 0.228$ | $97.953 \pm 0.182$ |
| | RNN | $99.953 \pm 0.023$ | $97.657 \pm 0.720$ | $97.813 \pm 0.676$ |
| | RNNSig | $99.814 \pm 0.044$ | $93.268 \pm 0.730$ | $93.747 \pm 0.743$ |
| | Sig | $96.736 \pm 0.578$ | $71.393 \pm 3.784$ | $73.245 \pm 3.642$ |
| forward-filling | DeepSig | $99.965 \pm 0.030$ | $97.974 \pm 0.381$ | $98.120 \pm 0.365$ |
| | RNN | $99.954 \pm 0.010$ | $97.786 \pm 0.308$ | $97.939 \pm 0.281$ |
| | RNNSig | $99.840 \pm 0.047$ | $94.110 \pm 0.774$ | $94.596 \pm 0.745$ |
| | Sig | $54.308 \pm 4.187$ | $7.387 \pm 2.995$ | $7.117 \pm 2.417$ |
| indicator | DeepSig | $99.955 \pm 0.033$ | $\mathbf{98.626 \pm 0.500}$ | $\mathbf{98.733 \pm 0.481}$ |
| | RNN | $99.953 \pm 0.024$ | $97.502 \pm 0.527$ | $97.660 \pm 0.499$ |
| | RNNSig | $99.755 \pm 0.078$ | $93.091 \pm 1.056$ | $93.635 \pm 0.952$ |
| | Sig | $66.917 \pm 18.306$ | $18.481 \pm 18.692$ | $19.067 \pm 19.165$ |
| linear | DeepSig | $\mathbf{99.984 \pm 0.007}$ | $\mathbf{98.898 \pm 0.205}$ | $\mathbf{98.997 \pm 0.201}$ |
| | RNN | $99.928 \pm 0.043$ | $97.668 \pm 0.897$ | $97.786 \pm 0.802$ |
| | RNNSig | $99.767 \pm 0.037$ | $92.754 \pm 0.662$ | $93.273 \pm 0.656$ |
| | Sig | $55.023 \pm 6.655$ | $9.436 \pm 3.349$ | $9.958 \pm 4.097$ |
| zero | DeepSig | $\mathbf{99.980 \pm 0.013}$ | $98.337 \pm 0.644$ | $98.454 \pm 0.616$ |
| | RNN | $99.887 \pm 0.052$ | $96.004 \pm 1.074$ | $96.253 \pm 1.046$ |
| | RNNSig | $99.685 \pm 0.063$ | $92.154 \pm 0.878$ | $92.744 \pm 0.820$ |
| | Sig | $96.997 \pm 0.388$ | $69.963 \pm 4.208$ | $71.699 \pm 4.002$ |

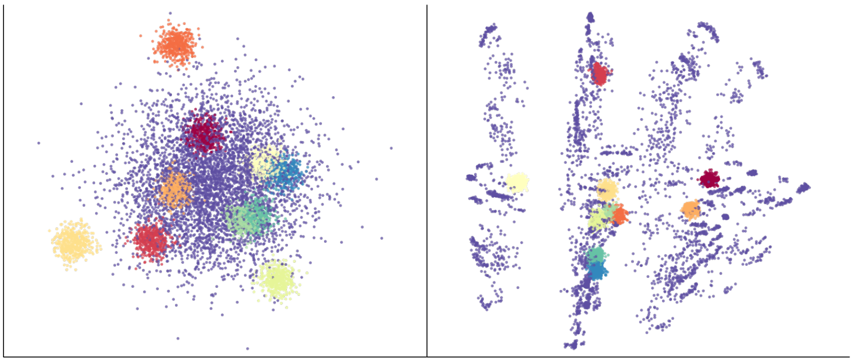
Table A.2: PenDigits, under random subsampling, $p = 30\%$

| metric | w-AUROC | BAC | Accuracy | |
|---|---|---|---|---|
| GP-PoM | DeepSig | $99.515 \pm 0.078$ | $92.151 \pm 0.555$ | $92.098 \pm 0.548$ |
| | RNN | $99.564 \pm 0.072$ | $\underline{92.757 \pm 0.735}$ | $\underline{92.699 \pm 0.733}$ |
| | RNNSig | $98.967 \pm 0.253$ | $88.148 \pm 1.588$ | $88.113 \pm 1.579$ |
| | Sig | $99.028 \pm 0.099$ | $87.352 \pm 0.898$ | $87.290 \pm 0.903$ |
| GP | DeepSig | $90.509 \pm 0.164$ | $54.545 \pm 0.426$ | $54.513 \pm 0.451$ |
| | RNN | $91.961 \pm 0.856$ | $57.930 \pm 2.079$ | $57.900 \pm 2.088$ |
| | RNNSig | $86.740 \pm 0.585$ | $46.842 \pm 1.255$ | $46.867 \pm 1.218$ |
| | Sig | $83.511 \pm 0.485$ | $41.747 \pm 0.428$ | $41.809 \pm 0.425$ |
| causal | DeepSig | $99.096 \pm 0.116$ | $89.480 \pm 0.359$ | $89.434 \pm 0.362$ |
| | RNN | $99.288 \pm 0.066$ | $89.526 \pm 0.535$ | $89.474 \pm 0.539$ |
| | RNNSig | $99.165 \pm 0.067$ | $88.807 \pm 0.613$ | $88.759 \pm 0.617$ |
| | Sig | $97.870 \pm 0.224$ | $80.065 \pm 0.980$ | $80.011 \pm 0.971$ |
| forward-filling | DeepSig | $99.141 \pm 0.068$ | $88.974 \pm 0.656$ | $88.902 \pm 0.644$ |
| | RNN | $99.311 \pm 0.067$ | $90.067 \pm 0.247$ | $90.029 \pm 0.247$ |
| | RNNSig | $99.203 \pm 0.063$ | $88.930 \pm 0.513$ | $88.902 \pm 0.528$ |
| | Sig | $98.425 \pm 0.069$ | $84.458 \pm 0.468$ | $84.374 \pm 0.477$ |
| indicator | DeepSig | $\mathbf{99.607 \pm 0.059}$ | $\mathbf{93.156 \pm 0.738}$ | $\mathbf{93.087 \pm 0.751}$ |
| | RNN | $\underline{\mathbf{99.733 \pm 0.044}}$ | $\underline{\mathbf{94.124 \pm 0.412}}$ | $\underline{\mathbf{94.071 \pm 0.415}}$ |
| | RNNSig | $99.549 \pm 0.041$ | $91.604 \pm 0.278$ | $91.532 \pm 0.268$ |
| | Sig | $98.708 \pm 0.040$ | $84.544 \pm 0.538$ | $84.505 \pm 0.563$ |
| linear | DeepSig | $99.407 \pm 0.151$ | $91.418 \pm 1.075$ | $91.366 \pm 1.086$ |
| | RNN | $99.510 \pm 0.041$ | $91.862 \pm 0.582$ | $91.812 \pm 0.594$ |
| | RNNSig | $\underline{99.591 \pm 0.036}$ | $91.556 \pm 0.518$ | $91.521 \pm 0.539$ |
| | Sig | $99.029 \pm 0.094$ | $87.116 \pm 0.612$ | $87.038 \pm 0.612$ |
| zero | DeepSig | $99.334 \pm 0.077$ | $89.774 \pm 0.541$ | $89.686 \pm 0.553$ |
| | RNN | $99.403 \pm 0.112$ | $90.729 \pm 0.618$ | $90.698 \pm 0.620$ |
| | RNNSig | $99.150 \pm 0.046$ | $87.948 \pm 0.248$ | $87.879 \pm 0.243$ |
| | Sig | $98.623 \pm 0.073$ | $83.935 \pm 0.382$ | $83.905 \pm 0.375$ |

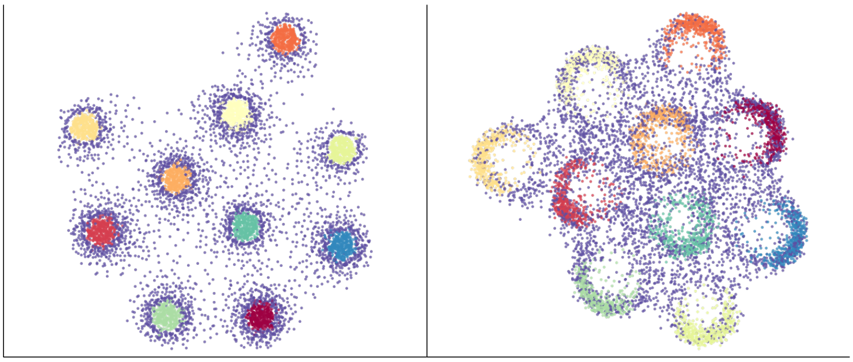
## A.3 TOPOLOGICAL REPRESENTATION LEARNING

In Figure A.20, the two-dimensional visualisations of the Spheres dataset are shown for all the included methods. In addition, we depict two further approaches. First, as an ablation of TopoAE, we apply our topological constraint to a linear autoencoder (TopoPCA) that has a single hidden layer of two dimensions. We find that also TopoPCA was able to preserve the manifold structure of the nested spheres to a certain degree, however less distinctively than TopoAE. Second, we applied the PHATE method to the Spheres dataset [137]. For this, during the hyperparameter search we varied the parameter knn between $5$ and $30$ (in $20$ random calls, same as the other methods) and otherwise used the default parameters: decay $= 40$, $\gamma = 1$, knn_dist $=$ euclidean, mds $=$ metric, mds_dist $=$ euclidean, mds_solver $=$ sgd, n_components $= 2$, n_jobs $= 1$, n_landmark $= 2000$, n_pca $= 100$, $t =$ auto. In the hyperparameter search, knn $= 5$ was found to perform best (in terms of $\mathrm{KL}_{0.1}$, the quantity we sought to minimise for all methods). Here, we consistently observed the displayed triangular pattern, where the surrounding sphere was severed, and multiple inner spheres were placed at the outer border of the embedding. This pattern is most similar to the embeddings of the classical AE.

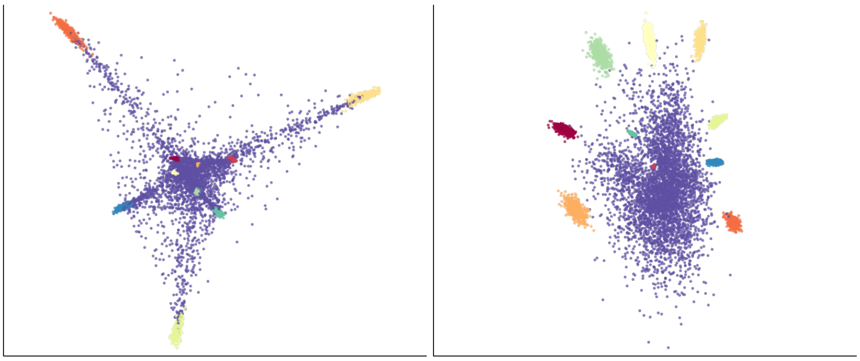
**a)** PCA **b)** Isomap



**c)** t-SNE **d)** UMAP



**e)** PHATE **f)** AE



**g)** TopoPCA **h)** TopoAE

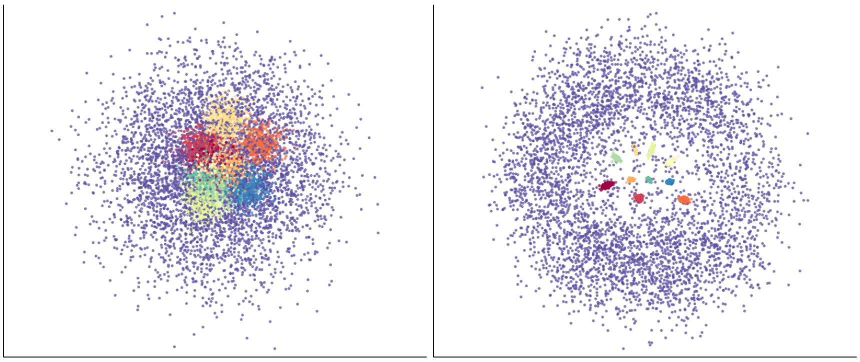


Figure A.20: Visualisations of the two-dimensional latent embeddings of the Spheres dataset. We observe that only our method, TopoAE was able to accurately capture the nested configuration of the sphere manifolds. Additionally, we add our topological regularisation term to a variant of PCA as implemented as a linear autoencoder (TopoPCA), which also preserves the nesting structure, albeit less clear than TopoAE. Finally, we consider an additional comparison partner, PHATE [137], which was published in the same year as TopoAE. We find that PHATE does not preserve the nesting relation of the spheres.

# ACRONYMS

| | |
|---|---|
| $k$NN | $k$-nearest neighbours |
| ABX | Antibiotics |
| AE | Autoencoder |
| attn | Self-attention model |
| AUMC | Amsterdam University Medical Centers Database |
| AUPRC | Area under the precision-recall curve |
| AUROC | Area under the receiver-operating-characteristic curve |
| BAC | Balanced accuracy |
| BCE | Binary cross-entropy |
| CI | Confidence interval |
| CNN | Convolutional neural network |
| CV | Cross-validation |
| DL | Deep learning |
| DTW | Dynamic time warping |
| DTW-$k$NN | Dynamic time warping $k$-nearest neighbour |
| EHR | Electronic health record |
| GCS | Glasgow Coma Scale |
| GP | Gaussian process |
| GP-PoM | GP adapter with posterior moments |
| gru | Gated recurrent units |
| HiRID | High time resolution ICU dataset |
| ICD-9 | International Classification of Diseases, Ninth Revision |
| ICU | Intensive care unit |
| IT | Information technology |
| lgbm | Light gradient boosting machine |
| lr | Logistic regression |
| LSTM | Long short-term memory network |
| MAP | Mean arterial blood pressure |
| MC | Monte Carlo |

| | |
|---|---|
| MDS | Multi-dimensional scaling |
| MEWS | Modified Early Warning Score |
| MGP | Multi-task Gaussian process |
| MGP-RNN | Gaussian process recurrent neural network |
| MGP-TCN | Gaussian process temporal convolutional network |
| MIMIC-III | Multiparameter Intelligent Monitoring in Intensive Care |
| ML | Machine learning |
| MLP | Multilayer perceptron |
| NEWS | National Early Warning Score |
| PCA | Principal component analysis |
| PE | Positional encoding |
| PH | Persistent homology |
| PPV | Positive predictive value |
| qSOFA | Quick SOFA |
| ReLU | Rectified linear unit |
| RNN | Recurrent neural network |
| ROC | Receiver operating characteristic |
| SD | Standard deviation |
| SI | Suspected infection |
| SIRS | Systemic Inflammation Response Syndrome |
| SOFA | Sequential Organ Failure Assessment |
| SVM | Support vector machine |
| t-SNE | t-distributed stochastic neighbour embedding |
| TCN | Temporal convolutional network |
| TDA | Topological data analysis |
| TopoAE | Topological autoencoder |
| TREWScore | Targeted real-time warning score |
| UMAP | Uniform manifold approximation and projection |
| VR | Vietoris–Rips complex |
| w-AUROC | Weighted AUROC |
| WTK | Wasserstein time series kernel |

# BIBLIOGRAPHY

1. H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. “Persistence Images: A Stable Vector Representation of Persistent Homology”. *Journal of Machine Learning Research* 18:1, 2017, pp. 218–252.
2. C. C. Aggarwal. *Data mining: the textbook*. Springer, 2015.
3. A. Arévalo, J. Niño, G. Hernández, and J. Sandoval. “High-frequency trading strategy based on deep neural networks”. In: *International Conference on Intelligent Computing*. Springer. 2016, pp. 424–436.
4. I. P. Arribas. “Derivatives pricing using signature payoffs”. *arXiv preprint arXiv:1809.09466*, 2018.
5. I. P. Arribas, G. M. Goodwin, J. R. Geddes, T. Lyons, and K. E. Saunders. “A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder”. *Translational psychiatry* 8:1, 2018, pp. 1–7.
6. J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer normalization”. *arXiv preprint arXiv:1607.06450*, 2016.
7. S. Bai, J. Z. Kolter, and V. Koltun. “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”. *arXiv preprint arXiv:1803.01271*, 2018.
8. D. L. Bailey, M. N. Maisey, D. W. Townsend, and P. E. Valk. *Positron emission tomography*. Vol. 2. Springer, 2005.
9. S. Barannikov. “The framed Morse complex and its invariants”. *American Mathematical Society Translations, Series 2*, 1994.
10. Y. Bengio, A. Courville, and P. Vincent. “Representation learning: A review and new perspectives”. *IEEE transactions on pattern analysis and machine intelligence* 35:8, 2013, pp. 1798–1828.
11. D. J. Berndt and J. Clifford. “Using dynamic time warping to find patterns in time series.” In: *KDD workshop*. Vol. 10. 16. Seattle, WA, USA: 1994, pp. 359–370.

12. S. Bhartiya, D. Mehrotra, and A. Girdhar. "Issues in achieving complete interoperability while sharing electronic health records". *Procedia Computer Science* 78, 2016, pp. 192–198.

13. 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. ISSN: 1367-4803. DOI: `10.1093/bioinformatics/bty246`.

14. C. Bock, M. Moor, C. R. Jutzeler, and K. M. Borgwardt. "Machine Learning for Biomedical Time Series Classification: From Shapelets to Deep Learning". In: *Artificial Neural Networks - Third Edition*. Ed. by H. M. Cartwright. Vol. 2190. Methods in Molecular Biology. Springer, 2021, pp. 33–71. DOI: `10.1007/978-1-0716-0826-5\_2`.

15. 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)*. IEEE. 2019, pp. 964–969.

16. A. Boles and P. Rad. "Voice biometrics: Deep learning-based voiceprint authentication system". In: *2017 12th System of Systems Engineering Conference (SoSE)*. IEEE. 2017, pp. 1–6.

17. 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.

18. E. V. Bonilla, K. M. A. Chai, and C. K. Williams. "Multi-task Gaussian Process prediction". In: *Advances in Neural Information Processing Systems*. 2007, pp. 153–160.

19. P. Bonnier, P. Kidger, I. P. Arribas, C. Salvi, and T. Lyons. "Deep signature transforms". In: *Advances in Neural Information Processing Systems*. 2019, pp. 3105–3115.

20. J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, A. Aujayeb, M. Moor, B. Rieck, and K. Borgwardt. "Accelerating detection of lung pathologies with explainable ultrasound image analysis". *Applied Sciences* 11:2, 2021, p. 672.

21. F. A. Bozza, J. I. Salluh, A. M. Japiassu, M. Soares, E. F. Assis, R. N. Gomes, M. T. Bozza, H. C. Castro-Faria-Neto, and P. T. Bozza. "Cytokine profiles as markers of disease severity in sepsis: a multiplex analysis". *Critical Care* 11:2, 2007, R49.

22. R. N. Bracewell and R. N. Bracewell. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986.

23. D. Burago, I. D. Burago, Y. Burago, S. Ivanov, S. V. Ivanov, and S. A. Ivanov. *A course in metric geometry*. Vol. 33. American Mathematical Soc., 2001.

24. J. S. Calvert, D. A. Price, U. K. Chettipally, C. W. Barton, M. D. Feldman, J. L. Hoffman, M. Jay, and R. Das. "A computational approach to early sepsis detection". *Computers in Biology and Medicine* 74, 2016, pp. 69–73.

25. S. D. Campbell and F. X. Diebold. "Weather forecasting for weather derivatives". *Journal of the American Statistical Association* 100:469, 2005, pp. 6–16.

26. M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, and Y. Umeda. "Perslay: A neural network layer for persistence diagrams and new graph topological signatures". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2786–2796.

27. M. Carrière, S. Y. Oudot, and M. Ovsjanikov. "Stable Topological Signatures for Points on 3D Shapes". In: *Proceedings of the Eurographics Symposium on Geometry Processing (SGP)*. Eurographics Association, Aire-la-Ville, Switzerland, 2015, pp. 1–12.

28. T. Chari, J. Banerjee, and L. Pachter. "The specious art of single-cell genomics". *bioRxiv*, 2021.

29. F. Chazal, D. Cohen-Steiner, and Q. Mérigot. "Geometric inference for probability measures". *Foundations of Computational Mathematics* 11:6, 2011, pp. 733–751.

30. F. Chazal, V. De Silva, and S. Oudot. "Persistence stability for geometric complexes". *Geometriae Dedicata* 173:1, 2014, pp. 193–214.

31. F. Chazal, B. Fasy, F. Lecci, B. Michel, A. Rinaldo, A. Rinaldo, and L. Wasserman. "Robust topological inference: Distance to a measure and kernel distance". *The Journal of Machine Learning Research* 18:1, 2017, pp. 5845–5884.

32. Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. "Recurrent neural networks for multivariate time series with missing values". *Scientific reports* 8:1, 2018, pp. 1–12.

33. C. Chen, X. Ni, Q. Bai, and Y. Wang. "A topological regularizer for classifiers via persistent homology". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2573–2582.

34. K.-T. Chen. "Integration of paths–A faithful representation of paths by noncommutative formal power series". *Transactions of the American Mathematical Society* 89:2, 1958, pp. 395–407.

35. K.-T. Chen. "Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula". *Annals of Mathematics*, 1957, pp. 163–178.

36. K.-T. Chen. “Iterated integrals and exponential homomorphisms”. *Proceedings of the London Mathematical Society* 3:1, 1954, pp. 502–512.
37. R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. “Neural ordinary differential equations”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 6572–6583.
38. I. Chevyrev and A. Kormilitzin. “A primer on the signature method in machine learning”. *arXiv preprint arXiv:1603.03788*, 2016.
39. I. Chevyrev, V. Nanda, and H. Oberhauser. “Persistence paths and signature features in topological data analysis”. *IEEE transactions on pattern analysis and machine intelligence* 42:1, 2018, pp. 192–202.
40. K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. “On the properties of neural machine translation: Encoder-decoder approaches”. *arXiv preprint arXiv:1409.1259*, 2014.
41. A. Choudhary, M. Kerber, and S. Raghvendra. “Improved topological approximations by digitization”. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2675–2688.
42. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. “Stability of persistence diagrams”. *Discrete & computational geometry* 37:1, 2007, pp. 103–120.
43. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.
44. M. Das and S. K. Ghosh. “Data-driven approaches for meteorological time series prediction: a comparative study of the state-of-the-art computational intelligence techniques”. *Pattern Recognition Letters* 105, 2018, pp. 155–164.
45. H. A. Dau, D. F. Silva, F. Petitjean, G. Forestier, A. Bagnall, and E. Keogh. “Judicious setting of Dynamic Time Warping’s window width allows more accurate classification of time series”. In: *2017 IEEE International Conference on Big Data*. IEEE. 2017, pp. 917–922.
46. I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai, et al. “Federated learning for predicting clinical outcomes in patients with COVID-19”. *Nature medicine* 27:10, 2021, pp. 1735–1743.

47. R. P. Dellinger, M. M. Levy, A. Rhodes, D. Annane, H. Gerlach, S. M. Opal, J. E. Sevransky, C. L. Sprung, I. S. Douglas, R. Jaeschke, T. M. Osborn, M. E. Nunnally, S. R. Townsend, K. Reinhart, R. M. Kleinpell, D. C. Angus, C. S. Deutschman, F. R. Machado, G. D. Rubenfeld, S. A. Webb, R. J. Beale, J.-L. Vincent, and R. Moreno. "Surviving Sepsis Campaign: International Guidelines for Management of Severe Sepsis and Septic Shock 2012". *Critical Care Medicine* 41:2, 2013, pp. 580–637. DOI: `10.1097/CCM.0b013e31827e83af`.

48. T. Desautels, J. Calvert, J. Hoffman, M. Jay, Y. Kerem, L. Shieh, D. Shimabukuro, U. Chettipally, M. D. Feldman, C. Barton, et al. "Prediction of sepsis in the intensive care unit with minimal electronic health record data: A machine learning approach". *JMIR Medical Informatics* 4:3, 2016, e28.

49. B. Djulbegovic and G. H. Guyatt. "Progress in evidence-based medicine: a quarter century on". *The Lancet* 390:10092, 2017, pp. 415–423.

50. D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

51. H. Edelsbrunner, J. Harer, et al. "Persistent homology-a survey". *Contemporary mathematics* 453, 2008, pp. 257–282.

52. M. Fan, X. Zhang, H. Qiao, and B. Zhang. "Efficient isometric multi-manifold learning based on the self-organizing method". *Information Sciences* 345, 2016, pp. 325–339.

53. A. Fermanian. "Embedding and learning with signatures". *Computational Statistics & Data Analysis* 157, 2021, p. 107148.

54. R. Ferrer, I. Martin-Loeches, G. Phillips, T. M. Osborn, S. Townsend, R. P. Dellinger, A. Artigas, C. Schorr, and M. M. Levy. "Empiric antibiotic treatment reduces mortality in severe sepsis and septic shock from the first hour: results from a guideline-based performance improvement program". *Critical Care Medicine* 42:8, 2014, pp. 1749–1755.

55. M. Fey, J. E. Lenssen, F. Weichert, and H. Müller. "Splinecnn: Fast geometric deep learning with continuous b-spline kernels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 869–877.

56. L. M. Fleuren, T. L. Klausch, C. L. Zwager, L. J. Schoonmade, T. Guo, L. F. Roggeveen, E. L. Swart, A. R. Girbes, P. Thoral, A. Ercole, et al. "Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy". *Intensive Care Medicine*, 2020, pp. 1–18.

57. M. Fortin, J. Haggerty, S. Sanche, and J. Almirall. “Self-reported versus health administrative data: implications for assessing chronic illness burden in populations. A cross-sectional study”. *CMAJ open* 5:3, 2017, E729.

58. V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt. “Gp-vae: Deep probabilistic time series imputation”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1651–1661.

59. P. K. Friz and N. B. Victoir. *Multidimensional stochastic processes as rough paths: theory and applications*. Vol. 120. Cambridge University Press, 2010.

60. D. J. Funk, J. E. Parrillo, and A. Kumar. “Sepsis and septic shock: a history”. *Critical care clinics* 25:1, 2009, pp. 83–101.

61. J. Futoma, S. Hariharan, and K. Heller. “Learning to detect sepsis with a multitask Gaussian process RNN classifier”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1174–1182.

62. J. Futoma, S. Hariharan, K. Heller, M. Sendak, N. Brajer, M. Clement, A. Bedoya, and C. O’brien. “An improved multi-output gaussian process rnn with real-time validation for early sepsis detection”. In: *Machine Learning for Healthcare Conference*. PMLR. 2017, pp. 243–254.

63. J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. “GPyTorch: Blackbox matrix-matrix Gaussian Process inference with GPU acceleration”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 7576–7586.

64. R. L. Gardner, E. Cooper, J. Haskell, D. A. Harris, S. Poplau, P. J. Kroth, and M. Linzer. “Physician stress and burnout: the impact of health information technology”. *Journal of the American Medical Informatics Association* 26:2, 2019, pp. 106–114.

65. J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. “Convolutional sequence to sequence learning”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1243–1252.

66. A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

67. K. R. Genga and J. A. Russell. “Update of sepsis in the intensive care unit”. *Journal of innate immunity* 9:5, 2017, pp. 441–455.

68. S. Geroulanos and E. T. Douka. “Historical perspective of the word “sepsis””. *Intensive care medicine* 32:12, 2006, pp. 2077–2077.

69. K. Gilev, E. Yastrebova, D. Strokotov, M. Yurkin, N. Karmadonova, A. Chernyshev, V. Lomivorotov, and V. Maltsev. “Advanced consumable-free morphological analysis of intact red blood cells by a compact scanning flow cytometer”. *Cytometry Part A* 91:9, 2017, pp. 867–873.

70. A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals”. *circulation* 101:23, 2000, e215–e220.

71. Q. Gong, Y. Chen, X. He, Z. Zhuang, T. Wang, H. Huang, X. Wang, and X. Fu. “DeepScan: Exploiting deep learning for malicious account detection in location-based social networks”. *IEEE Communications Magazine* 56:11, 2018, pp. 21–27.

72. A. Gracia, S. González, V. Robles, and E. Menasalvas. “A methodology to compare dimensionality reduction algorithms in terms of loss of quality”. *Information Sciences* 270, 2014, pp. 1–27.

73. S. Gudmundsson, T. P. Runarsson, and S. Sigurdsson. “Support vector machines and dynamic time warping for time series”. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 2772–2776.

74. F. Gül, M. K. Arslantaş, İ. Cinel, and A. Kumar. “Changing definitions of sepsis”. *Turkish journal of anaesthesiology and reanimation* 45:3, 2017, p. 129.

75. E. Gultepe, J. P. Green, H. Nguyen, J. Adams, T. Albertson, and I. Tagkopoulos. “From vital signs to clinical outcomes for patients with sepsis: a machine learning basis for a clinical decision support system”. *Journal of the American Medical Informatics Association* 21:2, 2014, pp. 315–325.

76. 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`.

77. W. H. Guss and R. Salakhutdinov. “On characterizing the capacity of neural networks using algebraic topology”. *arXiv preprint arXiv:1802.04443*, 2018.

78. B. Hambly and T. Lyons. “Uniqueness for the signature of a path of bounded variation and the reduced path group”. *Annals of Mathematics*, 2010, pp. 109–167.

79. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

80. T. Head, L. Gilles, and I. Shcherbatyi. *`scikit-optimize`: v0.5.2*. 2018. DOI: `10.5281/zenodo.1207017`.

81. A. R. Heller, S. T. Mees, B. Lauterwald, C. Reeps, T. Koch, and J. Weitz. "Detection of deteriorating patients on surgical wards outside the ICU by an automated MEWS-based early warning system with paging functionality". *Annals of surgery* 271:1, 2020, pp. 100–105.

82. K. E. Henry, D. N. Hager, P. J. Pronovost, and S. Saria. "A targeted real-time early warning score (TREWScore) for septic shock". *Science Translational Medicine* 7:299, 2015.

83. F. Hensel, M. Moor, and B. Rieck. "A Survey of Topological Machine Learning Methods". *Frontiers in Artificial Intelligence* 4, 2021, p. 52.

84. G. E. Hinton. "Connectionist Learning Procedures". *Artif. Intell.* 40:1-3, 1989, pp. 185–234. DOI: `10.1016/0004-3702(89)90049-0`.

85. J. Hirschberg and C. D. Manning. "Advances in natural language processing". *Science* 349:6245, 2015, pp. 261–266.

86. S. Hochreiter and J. Schmidhuber. "Long short-term memory". *Neural computation* 9:8, 1997, pp. 1735–1780.

87. C. Hofer, F. Graf, B. Rieck, M. Niethammer, and R. Kwitt. "Graph filtration learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4314–4323.

88. C. Hofer, R. Kwitt, M. Niethammer, and M. Dixit. "Connectivity-optimized representation learning via persistent homology". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2751–2760.

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

90. D. Holden, J. Saito, T. Komura, and T. Joyce. "Learning motion manifolds with convolutional autoencoders". In: *SIGGRAPH Asia 2015 Technical Briefs*. 2015, pp. 1–4.

91. J. Holder. *Tracking Coronavirus Vaccinations Around the World*. 2021. URL: `https://www.nytimes.com/interactive/2021/world/covid-vaccinations-tracker.html` (visited on 11/03/2021).

92. M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt. "Topological graph neural networks". *arXiv preprint arXiv:2102.07835*, 2021.

93. M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt. “Set functions for time series”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4353–4363.

94. 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, 2016.

95. S. L. Hyland, M. Faltys, M. Hüser, X. Lyu, T. Gumbsch, C. Esteban, C. Bock, M. Horn, M. Moor, B. Rieck, M. Zimmermann, D. Bodenham, K. Borgwardt, G. Rätsch, and T. M. Merz. “Early prediction of circulatory failure in the intensive care unit using machine learning”. *Nature Medicine* 26:3, 2020, pp. 364–373.

96. S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 448–456.

97. L. Jing and Y. Tian. “Self-supervised visual feature learning with deep neural networks: A survey”. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

98. A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark. “MIMIC-III, a freely accessible critical care database”. *Scientific Data* 3, 2016.

99. A. E. 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 Medical Informatics Association* 25:1, 2018, pp. 32–39.

100. M. Jones. “NEWSDIG: The national early warning score development and implementation group”. *Clinical Medicine* 12:6, 2012, p. 501.

101. H. J. Kam and H. Y. Kim. “Learning representations for the early detection of sepsis with deep neural networks”. *Computers in biology and medicine* 89, 2017, pp. 248–255.

102. E. Karakike, E. J. Giamarellos-Bourboulis, M. Kyprianou, C. Fleischmann-Struzek, M. W. Pletz, M. G. Netea, K. Reinhart, and E. Kyriazopoulou. “Coronavirus disease 2019 as cause of viral sepsis: a systematic review and meta-analysis”. *Critical care medicine*, 2021.

103. K. M. Kaukonen, M. Bailey, S. Suzuki, D. Pilcher, and R. Bellomo. “Mortality related to severe sepsis and septic shock among critically ill patients in Australia and New Zealand, 2000-2012.” *JAMA* 311:13, 2014, pp. 1308–1316.

104. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. “Lightgbm: A highly efficient gradient boosting decision tree”. *Advances in Neural Information Processing Systems* 30, 2017, pp. 3146–3154.

105. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. "Dimensionality reduction for fast similarity search in large time series databases". *Knowledge and information Systems* 3:3, 2001, pp. 263–286.
106. E. J. Keogh and M. J. Pazzani. "Scaling up dynamic time warping to massive datasets". In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 1999, pp. 1–11.
107. P. H. Le-Khac, G. Healy, and A. F. Smeaton. "Contrastive representation learning: A framework and review". *IEEE Access*, 2020.
108. 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.
109. P. Kidger and T. J. Lyons. "Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU". In: *International Conference on Learning Representations*. OpenReview.net, 2021. URL: `https://openreview.net/forum?id=lqU2cs3Zca`.
110. P. Kidger, J. Morrill, J. Foster, and T. J. Lyons. "Neural Controlled Differential Equations for Irregular Time Series". In: *Advances in Neural Information Processing Systems*. 2020.
111. J. Kim, A. S. Campbell, B. E.-F. de Ávila, and J. Wang. "Wearable biosensors for healthcare monitoring". *Nature biotechnology* 37:4, 2019, pp. 389–406.
112. D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: `http://arxiv.org/abs/1412.6980`.
113. F. J. Király and H. Oberhauser. "Kernels for sequentially ordered data". *Journal of Machine Learning Research* 20:31, 2019, pp. 1–45.
114. A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel. "Poincaré maps for analyzing complex hierarchies in single-cell data". *Nature communications* 11:1, 2020, pp. 1–9.
115. A. Kormilitzin, K. Saunders, P. Harrison, J. Geddes, and T. Lyons. "Application of the signature method to pattern recognition in the cequel clinical trial". *arXiv preprint arXiv:1606.02074*, 2016.
116. K.-M. Lau and H. Weng. "Climate signal detection using wavelet transform: How to make a time series sing". *Bulletin of the American meteorological society* 76:12, 1995, pp. 2391–2402.

117. C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. “Temporal convolutional networks for action segmentation and detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 156–165.

118. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation applied to handwritten zip code recognition”. *Neural computation* 1:4, 1989, pp. 541–551.

119. C. Lee. *Implementing Temporal Convolutional Networks*. 2018. URL: `https://colab.research.google.com/drive/1la33lW7FQV1RicpfzyLq9H0SH1VSD4LE`.

120. J. A. Lee and M. Verleysen. “Quality assessment of dimensionality reduction: Rank-based criteria”. *Neurocomputing* 72:7-9, 2009, pp. 1431–1443.

121. D. Levin, T. Lyons, and H. Ni. “Learning from the past, predicting the statistics for the future, learning an evolving system”. *arXiv preprint arXiv:1309.0260*, 2013.

122. M. M. Levy, M. P. Fink, J. C. Marshall, E. Abraham, D. Angus, D. Cook, J. Cohen, S. M. Opal, J.-L. Vincent, G. Ramsay, et al. “2001 SCCM/ESICM/ACCP/ATS/SIS International Sepsis Definitions Conference”. *Intensive Care Medicine* 29:4, 2003, pp. 530–538.

123. R. Lewis and D. Morozov. “Parallel computation of persistent homology using the blowup complex”. In: *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*. 2015, pp. 323–331.

124. C. Li, X. Zhang, and L. Jin. “LPSNet: a novel log path signature feature based hand gesture recognition framework”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 631–639.

125. S. C.-X. Li and B. M. Marlin. “A scalable end-to-end gaussian process adapter for irregularly sampled time series classification”. In: *Advances In Neural Information Processing Systems*. 2016, pp. 1804–1812.

126. S. Liao, T. Lyons, W. Yang, and H. Ni. “Learning stochastic differential equations using RNN with log signature features”. *arXiv preprint arXiv:1908.08286*, 2019.

127. D. Lin, J. Xiong, C. Liu, L. Zhao, Z. Li, S. Yu, X. Wu, Z. Ge, X. Hu, B. Wang, et al. “Application of Comprehensive Artificial intelligence Retinal Expert (CARE) system: a national real-world evidence study”. *The Lancet Digital Health* 3:8, 2021, e486–e495.

128. S. Linnainmaa. “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors”. *Master’s Thesis (in Finnish), Univ. Helsinki*, 1970, pp. 6–7.

129. 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. Curran Associates, Inc., 2017, pp. 4765–4774.
130. T. Lyons. "Rough paths, signatures and the modelling of functions on streams". *arXiv preprint arXiv:1405.4537*, 2014.
131. T. J. Lyons. "Differential equations driven by rough signals". *Revista Matemática Iberoamericana* 14:2, 1998, pp. 215–310.
132. L. J. van der Maaten and G. Hinton. "Visualizing data using t-SNE". *Journal of Machine Learning Research* 9, 2008, pp. 2579–2605.
133. S. Majumdar and A. K. Laha. "Clustering and classification of time series using topological data analysis with applications to finance". *Expert Systems with Applications* 162, 2020, p. 113868.
134. P. E. Marik and J. D. Farkas. "The changing paradigm of sepsis: early diagnosis, early antibiotics, early pressors, and early adjuvant treatment". *Critical care medicine* 46:10, 2018, pp. 1690–1692.
135. L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". *arXiv preprint arXiv:1802.03426*, 2018.
136. Y. Meyer. *Wavelets and Operators: Volume 1*. 37. Cambridge university press, 1992.
137. K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. van den Elzen, M. J. Hirn, R. R. Coifman, et al. "Visualizing structure and transitions in high-dimensional biological data". *Nature biotechnology* 37:12, 2019, pp. 1482–1492.
138. M. Moor, N. Bennett, D. Plečko, M. Horn, B. Rieck, N. Meinshausen, P. Bühlmann, and K. Borgwardt. "Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning". *arXiv preprint arXiv:2107.05230*, 2021.
139. M. Moor, M. Horn, C. Bock, K. Borgwardt, and B. Rieck. "Path Imputation Strategies for Signature Models". In: *ICML Workshop on the Art of Learning with Missing Values*. 2020.
140. M. Moor, M. Horn, K. Borgwardt, and B. Rieck. "Challenging euclidean topological autoencoders", 2020.
141. M. Moor, M. Horn, B. Rieck, and K. Borgwardt. "Topological Autoencoders". In: *International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054.

142. M. Moor, M. Horn, B. Rieck, D. Roqueiro, and K. Borgwardt. “Early recognition of sepsis with Gaussian process temporal convolutional networks and dynamic time warping”. In: *Machine Learning for Healthcare Conference*. PMLR. 2019, pp. 2–26.

143. 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. DOI: `10.3389/fmed.2021.607952`.

144. J. Morlet. “Sampling theory and wave propagation”. In: *Issues in acoustic Signal—image processing and recognition*. Springer, 1983, pp. 233–261.

145. J. Morrill, A. Kormilitzin, A. Nevado-Holgado, S. Swaminathan, S. Howison, and T. Lyons. “The signature-based model for early detection of sepsis from electronic health records in the intensive care unit”. In: *2019 Computing in Cardiology (CinC)*. IEEE. 2019, Page–1.

146. R. Müller, S. Kornblith, and G. E. Hinton. “When does label smoothing help?” In: *Advances in Neural Information Processing Systems*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett. 2019, pp. 4696–4705.

147. D. Nguyen-Tuong, J. Peters, and M. Seeger. “Local gaussian process regression for real time online model learning and control”. In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. 2008, pp. 1193–1200.

148. Z. Obermeyer and E. J. Emanuel. “Predicting the future—big data, machine learning, and clinical medicine”. *The New England journal of medicine* 375:13, 2016, p. 1216.

149. D. Oglic and T. Gärtner. “Learning in reproducing kernel Kreĭn spaces”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3859–3867.

150. E. Oja, K. Kiviluoto, and S. Malaroiu. “Independent component analysis for financial time series”. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. IEEE. 2000, pp. 111–116.

151. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. “WaveNet: A Generative Model for Raw Audio”. In: *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*. ISCA, 2016, p. 125. URL: `http://www.isca-speech.org/archive/SSW%5C_2016/abstracts/ssw9%5C_DS-4%5C_van%5C_den%5C_Oord.html`.

152. I. Oregi, A. Pérez, J. Del Ser, and J. A. Lozano. “On-line dynamic time warping for streaming time series”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2017, pp. 591–605.

153. E. Otles, J. Oh, B. Li, M. Bochinski, H. Joo, J. Ortwine, E. Shenoy, L. Washer, V. B. Young, K. Rao, et al. “Mind the Performance Gap: Examining Dataset Shift During Prospective Validation”. In: *Machine Learning for Healthcare Conference*. PMLR. 2021, pp. 506–534.

154. R. Paul and S. K. Chalup. “A study on validating non-linear dimensionality reduction using persistent homology”. *Pattern Recognition Letters* 100, 2017, pp. 160–166.

155. S. A. Pendergrass and D. C. Crawford. “Using electronic health records to generate phenotypes for research”. *Current protocols in human genetics* 100:1, 2019, e80.

156. G. Peyré. “Manifold models for signals and images”. *Computer vision and image understanding* 113:2, 2009, pp. 249–260.

157. L. X. Polastron. *Books on fire: The destruction of libraries throughout history*. Lucien X. POLASTRON, 2007.

158. T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi. “The eICU Collaborative Research Database, a freely available multi-center database for critical care research”. *Scientific Data* 5, 2018, p. 180178.

159. A. Poulenard, P. Skraba, and M. Ovsjanikov. “Topological Function Optimization for Continuous Shape Matching”. *Computer Graphics Forum* 37:5, 2018, pp. 13–25.

160. J. Quinonero-Candela and C. E. Rasmussen. “A unifying view of sparse approximate Gaussian process regression”. *The Journal of Machine Learning Research* 6, 2005, pp. 1939–1959.

161. K. N. Ramamurthy, K. Varshney, and K. Mody. “Topological data analysis of decision boundaries with application to model selection”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5351–5360.

162. C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN: 026218253X. URL: `https://www.worldcat.org/oclc/61285753`.

163. R. M. K. T. Rathnayaka, D. M. K. N. Seneviratna, J. Wei, and H. I. Arumawadu. “A hybrid statistical approach for stock market forecasting based on Artificial Neural Network and ARIMA time series models”. In: *2015 International Conference on Behavioral, Economic and Socio-cultural Computing, BESC 2015, Nanjing, China, October 30*

- *December 1, 2015*. Ed. by G. Xu, Y. Demazeau, S.-H. Chen, J. Wu, M. Gavin, I. King, J. Cao, Z. Wu, and Z. Bu. IEEE, 2015, pp. 54–60. DOI: `10.1109/BESC.2015.7365958`.

164. B. Reeder and A. David. "Health at hand: A systematic review of smart watch uses for health and wellness". *Journal of biomedical informatics* 63, 2016, pp. 269–276.

165. J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. "A stable multi-scale kernel for topological machine learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4741–4748.

166. M. A. Reyna, C. S. Josef, R. Jeter, S. P. Shashikumar, M. B. Westover, S. Nemati, G. D. Clifford, and A. Sharma. "Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019". *Critical Care Medicine* 48:2, 2020.

167. A. Rhodes, L. E. Evans, W. Alhazzani, M. M. Levy, M. Antonelli, R. Ferrer, A. Kumar, J. E. Sevransky, C. L. Sprung, M. E. Nunnally, et al. "Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016". *Intensive Care Medicine* 43:3, 2017, pp. 304–377.

168. V. J. Ribas, J. C. López, A. Ruiz-Sanmartín, J. C. Ruiz-Rodríguez, J. Rello, A. Wojdel, and A. Vellido. "Severe sepsis mortality prediction with relevance vector machines". In: *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2011*. IEEE, 2011, pp. 100–103. DOI: `10.1109/IEMBS.2011.6089906`.

169. B. Rieck. "Persistent Homology in Multivariate Data Visualization". PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2017. DOI: `10.11588/heidok.00022914`.

170. B. Rieck and H. Leitte. "Agreement Analysis of Quality Measures for Dimensionality Reduction. Theory, Algorithms, and Applications". In: *Topological Methods in Data Analysis and Visualization IV*. Ed. by H. Carr, C. Garth, and T. Weinkauf. Springer, Cham, Switzerland, 2017.

171. B. Rieck and H. Leitte. "Persistent Homology for the Evaluation of Dimensionality Reduction Schemes". *Computer Graphics Forum* 34:3, 2015, pp. 431–440.

172. 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*. 2019.

173. N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al. "The future of digital health with federated learning". *NPJ digital medicine* 3:1, 2020, pp. 1–7.

174. M. C. van Rossum, L. B. Vlaskamp, L. M. Posthuma, M. J. Visscher, M. J. Breteler, H. J. Hermens, C. J. Kalkman, and B. Preckel. “Adaptive threshold-based alarm strategies for continuous vital signs monitoring”. *Journal of clinical monitoring and computing*, 2021, pp. 1–11.

175. Y. Rubanova, R. T. Chen, and D. Duvenaud. “Latent ODEs for irregularly-sampled time series”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 5320–5330.

176. D. B. Rubin. “Inference and missing data”. *Biometrika* 63:3, 1976, pp. 581–592.

177. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. *nature* 323:6088, 1986, pp. 533–536.

178. T. Saito and M. Rehmsmeier. “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets”. *PloS one* 10:3, 2015, e0118432.

179. T. Salimans and D. P. Kingma. “Weight normalization: A simple reparameterization to accelerate training of deep neural networks”. *Advances in neural information processing systems* 29, 2016, pp. 901–909.

180. C. Salvi, T. Cass, J. Foster, T. Lyons, and W. Yang. “The Signature Kernel is the solution of a Goursat PDE”. *SIAM Journal on Mathematics of Data Science* 3:3, 2021, pp. 873–899.

181. C. W. Seymour, V. X. Liu, T. J. Iwashyna, F. M. Brunkhorst, T. D. Rea, A. Scherag, G. Rubenfeld, J. M. Kahn, M. Shankar-Hari, M. Singer, C. S. Deutschman, G. J. Escobar, and D. C. Angus. “Assessment of Clinical Criteria for Sepsis: For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)”. *JAMA* 315:8, 2016, pp. 762–774.

182. E. Sherman, H. Gurm, U. Balis, S. Owens, and J. Wiens. “Leveraging clinical time-series data for prediction: a cautionary tale”. In: *AMIA Annual Symposium Proceedings*. Vol. 2017. American Medical Informatics Association. 2017, p. 1571.

183. D. W. Shimabukuro, C. W. Barton, M. D. Feldman, S. J. Mataraso, and R. Das. “Effect of a machine learning-based severe sepsis prediction algorithm on patient survival and hospital length of stay: a randomised clinical trial”. *BMJ open respiratory research* 4:1, 2017, e000234.

184. S. N. Shukla and B. M. Marlin. “Interpolation-Prediction Networks for Irregularly Sampled Time Series”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

185. 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.
186. Q. Skinner. *Machiavelli: the prince*. 1988.
187. E. Snelson and Z. Ghahramani. "Sparse Gaussian processes using pseudo-inputs". *Advances in Neural Information Processing Systems* 18, 2006, p. 1257.
188. A. Solin et al. "Stochastic differential equation methods for spatio-temporal Gaussian process regression", 2016.
189. H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias. "Attend and diagnose: Clinical time series analysis using attention models". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.
190. C. Subbe, M. Kruger, P. Rutherford, and L. Gemmel. "Validation of a modified Early Warning Score in medical admissions". *QJM* 94:10, 2001, pp. 521–526.
191. M. Sundararajan, A. Taly, and Q. Yan. "Axiomatic Attribution for Deep Networks". In: *International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328.
192. B. Szubert, J. E. Cole, C. Monaco, and I. Drozdov. "Structure-preserving visualisation of high dimensional single-cell datasets". *Scientific reports* 9:1, 2019, pp. 1–10.
193. Y. Tashiro, J. Song, Y. Song, and S. Ermon. "CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation". *arXiv preprint arXiv:2107.03502*, 2021.
194. J. B. Tenenbaum, V. De Silva, and J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction". *Science* 290:5500, 2000, pp. 2319–2323.
195. M. Thill, W. Konen, and T. Bäck. "Time Series Encodings with Temporal Convolutional Networks". In: *International Conference on Bioinspired Methods and Their Applications*. Springer. 2020, pp. 161–173.
196. P. J. Thoral, J. M. Peppink, R. H. Driessen, E. J. Sijbrands, E. J. Kompanje, L. Kaplan, H. Bailey, J. Kesecioglu, M. Cecconi, M. Churpek, et al. "Sharing ICU patient data responsibly under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) example". *Critical Care Medicine*, 2021.

197. R. Tibshirani. “Regression Shrinkage and Selection via the Lasso”. *Journal of the Royal Statistical Society, Series B (Methodological)* 58:1, 1996, pp. 267–288.

198. E. Topol. *Deep medicine: how artificial intelligence can make healthcare human again*. Hachette UK, 2019.

199. C. Toth and H. Oberhauser. “Bayesian learning from sequential data using gaussian processes with signature covariances”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9548–9560.

200. E. L. Tsalik, L. B. Jaggers, S. W. Glickman, R. J. Langley, J. C. van Velkinburgh, L. P. Park, V. G. Fowler, C. B. Cairns, S. F. Kingsmore, and C. W. Woods. “Discriminative value of inflammatory biomarkers for suspected sepsis”. *The Journal of Emergency Medicine* 43:1, 2012, pp. 97–106.

201. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

202. J. Venna and S. Kaski. “Visualizing gene interaction graphs with local multidimensional scaling.” In: *ESANN*. Vol. 6. Citeseer. 2006, pp. 557–562.

203. L. Vietoris. “Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen”. *Mathematische Annalen* 97:1, 1927, pp. 454–472.

204. J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, and L. Thijs. “The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure”. *Intensive Care Medicine* 22:7, 1996, pp. 707–710.

205. H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2003.

206. H. Wagner and P. Dłotko. “Towards topological analysis of high-dimensional feature spaces”. *Computer Vision and Image Understanding* 121, 2014, pp. 21–26.

207. C. L. Walsh, P. Tafforeau, W. L. Wagner, D. J. Jafree, A. Bellier, C. Werlein, M. P. Kühnel, E. Boller, S. Walker-Samuel, J. L. Robertus, D. A. Long, J. Jacob, S. Marussi, E. Brown, N. Holroyd, D. D. Jonigk, M. Ackermann, and P. D. Lee. “Imaging intact human organs with local resolution of cellular structures using hierarchical phase-contrast tomography”. *Nature Methods*, 2021. DOI: `10.1038/s41592-021-01317-x`.

208. J. Wiens, J. Guttag, and E. Horvitz. “Patient risk stratification with time-varying parameters: a multitask learning approach”. *The Journal of Machine Learning Research* 17:1, 2016, pp. 2797–2819.

209. C. Williams, E. V. Bonilla, and K. M. Chai. “Multi-task Gaussian process prediction”. *Advances in Neural Information Processing Systems*, 2007, pp. 153–160.

210. A. Wilson and H. Nickisch. “Kernel interpolation for scalable structured Gaussian processes (KISS-GP)”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1775–1784.

211. A. Wong, E. Otles, J. P. Donnelly, A. Krumm, J. McCullough, O. DeTroyer-Cooley, J. Pestrue, M. Phillips, J. Konye, C. Penoza, et al. “External Validation of a Widely Implemented Proprietary Sepsis Prediction Model in Hospitalized Patients”. *JAMA Internal Medicine*, 2021.

212. Y.-L. Wu, D. Agrawal, and A. El Abbadi. “A comparison of DFT and DWT based similarity search in time-series databases”. In: *Proceedings of the ninth international conference on Information and knowledge management*. 2000, pp. 488–495.

213. Z. Wu, Y. Yang, Y. Ma, Y. Liu, R. Zhao, M. Moor, and V. Tresp. “Learning Individualized Treatment Rules with Estimated Translated Inverse Propensity Score”. In: *2020 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2020, pp. 1–11.

214. F. Xia and M. Yetisgen-Yildiz. “Clinical corpus annotation: challenges and strategies”. In: *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM’2012) in conjunction with the International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*. 2012, p. 67.

215. L. Yan, Y. Zhao, P. Rosen, C. Scheidegger, and B. Wang. “Homology-Preserving Dimensionality Reduction via Manifold Landmarking and Tearing”. *arXiv e-prints*, arXiv:1806.08460, 2018, arXiv:1806.08460. arXiv: `1806.08460 [cs.CG]`.

216. F. Yu and V. Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. 2016. URL: `http://arxiv.org/abs/1511.07122`.

217. S. Zeng, F. Graf, C. Hofer, and R. Kwitt. “Topological Attention for Time Series Forecasting”. *arXiv preprint arXiv:2107.09031*, 2021.

218. M. M. Zhang, B. Dumitrascu, S. A. Williamson, and B. E. Engelhardt. “Sequential Gaussian processes for online learning of nonstationary functions”. *arXiv preprint arXiv:1905.10003*, 2019.

219. S. Zhang, M. Xiao, and H. Wang. "GPU-Accelerated computation of Vietoris-Rips persistence barcodes". *arXiv preprint arXiv:2003.07989*, 2020.

220. Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. "Object detection with deep learning: A review". *IEEE transactions on neural networks and learning systems* 30:11, 2019, pp. 3212–3232.

221. J. Zhou and O. G. Troyanskaya. "An analytical framework for interpretable and generalizable single-cell data analysis". *Nature Methods*, 2021, pp. 1–5.

222. X. Zhu and A. B. Goldberg. "Introduction to semi-supervised learning". *Synthesis lectures on artificial intelligence and machine learning* 3:1, 2009, pp. 1–130.

223. A. Zomorodian. "Fast construction of the Vietoris-Rips complex". *Computers & Graphics* 34:3, 2010, pp. 263–271.