DOCTORAL THESIS

Predictive Systems Using Machine Learning Tools to Forecast Adverse Events During Medical Stays

Nzamba Bignoumba

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2024

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
49/2024

Predictive Systems Using Machine Learning Tools to Forecast Adverse Events During Medical Stays

NZAMBA BIGNOUMBA

TAL
TECH
PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

The dissertation was accepted for the defence of the degree of Doctor of Philosophy (Computer Science) on 2 September 2024

Supervisor: Professor Sadok Ben Yahia,
Department of Software Science,
School of Information Technologies,
Tallinn University of Technology,
Tallinn, Estonia

Co-supervisor: Professor Nedra Mellouli,
Department of Software Science,
ESILV, Léonrd de Vinci Group,
Paris, France

Opponents: Professor Vannary MEAS-YEDID HARDY,
Institut Pasteur,
Paris, France

Professor Henning Christiansen,
Roskilde University,
Roskilde, Denmark

Defence of the thesis: 27 September 2024, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Nzamba Bignoumba

signature

Copyright: Nzamba Bignoumba, 2024
ISSN 2585-6898 (publication)
ISBN 978-9916-80-194-9 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-195-6 (PDF)
DOI https://doi.org/10.23658/taltech.49/2024
Printed by Koopia Niini & Rauam

Nzamba Bignoumba (2024). *Predictive Systems Using Machine Learning Tools to Forecast Adverse Events During Medical Stays* [TalTech Press]. https://doi.org/10.23658/taltech.49/2024

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
49/2024

Ennustavad süsteemid, mis kasutavad masinõppe vahendeid kõrvalekallete prognoosimiseks haiglas viibimise ajal

NZAMBA BIGNOUMBA

KIRJASTUS

Contents

List of Publications 7

Author's Contributions to the Publications 8

Abbreviations 9

1 Introduction 10

2 Focus and Aim 13
  2.1 Research Questions 13

3 Research Methodology 16

4 Related Research 19
  4.1 Predictive Models Using Multivariate Clinical Time Series 19
  4.2 Predictive Models Using Longitudinal Patient Data 20

5 Publication-specific Contributions 21
  5.1 A New Efficient ALignment-driven Neural Network for Mortality Prediction from Irregular Multivariate Time Series data [I] 21
  5.2 Deep Padding and Alignment Strategies for Irregular Multivariate Clinical Time Series [II] 23
  5.3 Evaluation of Deep Learning-Based Depression Detection Using Medical Claims Data [III] 24
  5.4 Deep Magnitude Management of Clinical Code Embeddings to Predict Unplanned Hospital Readmissions [under review] 26

6 Discussion of Challenges 28

7 Conclusion 30

List of Figures 31

List of Tables 32

References 33

Acknowledgements 39

Abstract 40

Kokkuvõte 42

Appendix 1 45

Appendix 2 67

Appendix 3 79

Appendix 4 97

Curriculum Vitae 130

Elulookirjeldus 132

List of Publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

I N. Bignoumba, N. Mellouli, and S. B. Yahia. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. *Expert Systems with Applications*, 238:122148, 2024

II N. Bignoumba and S. Ben Yahiaand N. Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. In *Proceedings of KES'2024 - the 28th Annual KES Conference*, 2024

III M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, and D. Draheim. Evaluation of deep learning-based depression detection using medical claims data. *Artificial Intelligence in Medicine*, 147:102745, 2024

IV N. Bignoumba, M. Bertl, S. B. Yahia, and N. Mellouli. Deep magnitude management of clinical code embeddings to predict unplanned hospital readmissions. *PREPRINT (Version 2) available at Research Square*, 2024

Author's Contributions to the Publications

I In I, I was the main author, did the conceptualization, wrote the methodology, developed the model, carried out the validation and visualization, and wrote the manuscript.

II In II, I was the main author, did the conceptualization, wrote the methodology, developed the model, carried out the validation, and wrote the manuscript.

III In III, I was the second author, but I contributed equally with the first author. I did the conceptualization, conducted the survey, wrote the methodology, developed the model, performed the validation and visualization, and wrote the manuscript.

IV In IV, I was the main author, did the conceptualization, wrote the methodology, developed the model, carried out the validation and visualization, and wrote the manuscript.

Abbreviations

ALignment-driven Neural Network	ALNN
Area Under the Precision-Recall Curve	AUPRC
Area Under the ROC Curve	AUC
Artificial intelligence	AI
Convolutional Neural Networks	CNN
Electronic health record	EHR
Gated recurrent unit	GRU
Intensive care units	ICU
Machine learning	ML
Natural language processing	NLP
Ordinary differential equations	ODE
Recurrent Neural Network	RNN

1 Introduction

Advanced tools such as ChatGPT and Midjourney have recently drawn mainstream attention to the power of AI algorithms. However, Machine Learning (ML), which is the sub-field of AI behind these spectacular tools, is an old field. It was introduced between 1957 and 1960 [19] and has since benefited from advances in research in mathematics, electronics, algorithms, and software development, to name a few, making it a vital area in the advancement of various technologies. ML and its subfield, Deep Learning (DL), focus on developing algorithms that use data to mimic human decision-making. With their ability to learn hidden information contained in data that humans can not extract and quantify, ML algorithms have proven to be more efficient than human decision-making in a considerable number of real-world applications, such as image classification [11, 45], natural language processing (NLP) [56, 49] and stock market predictions [32, 34]. Due to its superior performance in a variety of real-world applications, it has attracted the attention of researchers and healthcare professionals.

The advancements in various fields of AI have somewhat alleviated the prejudices and fears that healthcare professionals previously held about this technology. The use of AI in medical research is expanding exponentially. This is largely driven by the adoption of Electronic Health Records (EHR) [23] in healthcare facilities. The application of ML algorithms on various complex tasks such as drug discovery [29], cancer detection [30], and mortality prediction [35], has proved highly effective. This effectiveness is proof that ML should be used on a large scale to help healthcare professionals in their decision-making. Beyond their predictive power, ML tools enable the delivery of better care, reduce the risk of practitioner burnout, improve patient health, better manage hospital resources, and reduce costs for both patients and hospitals, to name but a few [4, 53]. It is in this perspective of continuing to provide effective ML predictive tools that will improve the delivery of care and, consequently, the social lives of patients and healthcare professionals that this thesis is set.

Before the AI venue, medicine was more preventive and curative. With its arrival, researchers are trying to make it more predictive by drawing on the massive and heterogeneous quantity of data contained in EHRs. Compared with statistical models, ML models, particularly DL models, have a better ability to handle longitudinal data while preserving relevant information. As a result, they are ideal predictive models in medical settings, where patient health data are frequently recorded over time and across successive visits or admissions. Researchers have proposed several ML models trained on individual or mixed EHR data, including patient demographics, images, physiological measurements, and clinical codes, to predict or detect potential adverse medical events during a patient's medical stay. Despite achieving satisfactory results in various medical prediction tasks, these models still face certain limitations, namely suboptimal processing of irregular temporal data and suboptimal processing of historical medical data. In the literature [27, 2, 54], many models using data from successive patient admissions (or visits) often neglect the days elapsed between these admissions or assume that the days elapsed are uniform, meaning that the number of days between admissions is always the same. This is undoubtedly a flawed approach, as the level of importance the model attaches to recent data will be the same as that it attaches to older data. By exploiting the power of DL algorithms, as in previous related studies, this thesis seeks to address the above pitfalls and provide more accurate predictive models, whether for irregular temporal numerical data or irregular temporal categorical data.

To achieve the aim of the thesis, the author first identified the adverse events likely to occur during a patient's stay and the type of data that could be used to predict them. The author first hypothesizes that while there may be a multitude of events, such as sepsis or surgery, that may occur during the patient's stay, the worst is undoubtedly death. Therefore, mortality prediction is the primary concern of the thesis and irregular physiological time series measurements are used to carry out this prediction. The author bases the second hypothesis on the notion that an adverse medical event can be prevented by promptly identifying the illness that is likely to cause it. So, the author developed a deep learning model that uses historical and current medical data and patient demographics to detect eventual illnesses during a patient visit. Due to the area of expertise and data accessibility, the author chose depression detection as a pilot case. A recent survey by the National Health Interview (NIH) found that approximately 3.5% of deaths at the population level were linked to depression or anxiety [39]. These statistics reveal the crucial importance of detecting depression at an early stage. Finally, the author hypothesizes that predicting the patient's likelihood of readmission could potentially prevent adverse medical events. Some studies in the literature have highlighted the fact that unplanned readmissions worsen patients' state of health [31, 48]. Part of the thesis therefore focuses on predicting unplanned hospital readmission upon patient discharge. A deep learning model that leverages historical and current medical data, patient demographics and additional stay information was then developed to perform this prediction.

Through 4 articles (3 of which are published and one currently under review), this thesis presents 4 different deep-learning models that aim to improve the accuracy of predicting adverse medical events. Two of these models aim to predict mortality in Intensive Care Units (ICU), while the other two focus on detecting depression and predicting unplanned hospital readmissions. As demonstrated by extensive experiments conducted in publications, deep learning models are ideal and robust tools that can support physicians in their decision-making. The superior performance of the proposed models compared to state-of-the-art models highlights the effectiveness of the different strategies implemented to account for relevant aspects of the data. While the short-term goal of this thesis is to use machine learning tools to effectively predict adverse medical events, the long-term goal is to improve healthcare delivery and hospital management.

Despite their effectiveness, the proposed models have some limitations. These include a lack of explainability, a failure to account for uncertainty in the imputed values, and the use of a few clinical features as predictors. As these issues need to be resolved before the models can be deployed in real medical scenarios, future work will focus on overcoming the limitations faced by each model. In addition, the biases of the models and the ethical aspects associated with their use will be studied. The development of a unified system in which all proposed models work in synergy rather than individually will also be explored.

The thesis is carefully organized with a well-structured framework that includes several key chapters:

- **Chapter 2:** "Focus and Aim": This chapter presents the research questions and the various publications dealing with these questions.

- **Chapter 3:** "Research Methodology": This chapter describes the applied research methodology used to conduct the various studies carried out as part of the thesis.

- **Chapter 4:** “Related Research”: This chapter presents an overview of the literature on the prediction of adverse medical events from multivariate clinical time series and longitudinal patient health data [1].

- **Chapter 5:** “Publication-specific Contributions”: This chapter provides a high-level overview of the components of the models introduced in each article;

- **Chapter 6:** “Discussion of Challenges”: This chapter presents the challenges encountered during the research journey.

- **Chapter 7:** “Conclusion”: The final chapter provides a concise summary of the main findings and contributions of the thesis.

---

[1] Although multivariate clinical time series are also longitudinal data, throughout the thesis, longitudinal specifically refers to data collected over successive admissions or visits.

2 Focus and Aim

The short-term goal of this PhD thesis is to propose various deep-learning models that overcome underlying clinical data issues identified in the literature to provide accurate predictions of adverse events during the patient's medical stay. The long-term goal is to use these models in real-world medical scenarios to: assist physicians in their decision-making; relieve hospital congestion; reduce mortality rates; and improve the quality of care, to name a few. Having hypothesized that the worst event that might occur during a patient's stay is death and that some adverse medical events can be avoided by detecting in advance possible illnesses or predicting unplanned hospital readmissions, this thesis seeks to answer the research questions mentioned in Section 2.1.

2.1 Research Questions

- **RQ.1:** How to effectively predict mortality using irregular physiological time series measurements?
  This research question focuses on how to effectively predict mortality from physiological time series measurements that suffer from temporal irregularities, leading to underlying problems such as data sparsity and increased missing values. Temporal regularity refers to irregular time intervals between successive observations of a univariate time series. In the case of multivariate time series, irregularities may exhibit different patterns across univariate time series. Temporal irregularity is generally due to the fact that medical sensors have different and irregular data collection frequencies. Processing directly irregular physiological time series undoubtedly makes predictive machine learning or deep learning models suboptimal, according to the literature [13, 37]. For a task as sensitive as mortality, it is vital to propose a predictive model that overcomes the underlying problems associated with temporal irregularity. In summary, this research question aims to develop a model that optimally handles irregular temporal numerical (continuous) data, especially physiological measurements, to improve the accuracy of mortality prediction.

- **RQ.2:** How can adverse events be effectively predicted based on longitudinal data, i.e. historical and current patient data?
  The second research question aims to propose deep learning models that optimally process longitudinal patient data to prevent adverse events. It is observed in the literature that several models relying on historical and current patient data overlook the elapsed days between admissions (or visits) and their inherent irregularity (i.e., variation in elapsed days). This approach is neither optimal nor realistic, as it involves giving equal weight to historical and current medical information, leading to inaccurate downstream prediction. Furthermore, although some models in the literature [3, 14] implement attention mechanisms to focus on the most relevant clinical features, none of them have considered developing a mechanism that explicitly focuses on frequent medical events such as chronic diseases. Indeed, the latter are often the cause of some adverse events. Elapsed days and tracking frequent medical events are two key aspects that should be considered when using patient longitudinal data to predict adverse events. It is worth mentioning that, unlike the first research question, the data used in this research question are irregular temporal categorical data. Indeed, longitudinal data encompass categorical features such as diagnoses, procedures, and medication of consecutive admissions occurring at irregular time intervals, hence the term irregular temporal categorical data.

Four papers have proposed solutions to these research questions. These papers consist of building new deep-learning models or combining existing deep-learning architectures to predict adverse events during a patient's medical stay. Each article is based on one of the hypotheses put forward in the introduction. They contribute to answering the main research question, which is to predict adverse medical events during the patient's medical stay. These predictions are made using deep learning models capable of efficiently processing clinical data, solving their underlying problems and modelling temporal information.

- **Publication I "A new efficient ALignment-driven Neural Network for Mortality Prediction from Irregular Multivariate Time Series Data"** addresses partially the first research question. It introduces a new deep learning model called ALignment-driven Neural Network (ALNN) built on top of a Recurrent Neural Network (RNN) to make the mortality prediction task more accurate. RNN, the state-of-the-art model for time series analysis, has difficulty handling irregular multivariate time series [37], as it is intrinsically designed for regular time series. ALNN therefore provides it with a pseudo-regular version of irregular physiological time series called pseudo-aligned latent values. Extensive experiments have shown that ALNN makes RNNs, especially the Gated Recurrent Unit (GRU), more accurate when it comes to predicting mortality in ICU based on patients' physiological measurements.

- **Publication II "Deep Padding and Alignment Strategies for Irregular Multivariate Clinical Time Series"** is the sequel to publication $[I]$. It answers the first research question by integrating a data-driven imputation and padding approach in the prediction process. Various experiments were carried out to demonstrate the effectiveness of data-driven imputation to fill in missing values and data-driven padding to obtain univariate time series of equal length. This extended architecture has improved the accuracy of the mortality prediction task.

- **Publication III "Evaluation of Deep Learning-Based Depression Detection Using Medical Claims Data"** answers the second research question through a use case. This paper presents how the Self-Attention mechanism [49] can be combined with GRU-decay [12] to optimally detect depression from longitudinal patient claims data. The Self-Attention is used to model the relationship between diagnoses and filter out irrelevant diagnoses. While GRU-decay models temporal information. Compared to state-of-the-art models, the superior results of the proposed model demonstrate the critical importance of considering temporal information and accurately encoding relationships between diagnoses when detecting depression.

- **Under review: "Deep Magnitude Management of Clinical Code Embeddings to Predict Unplanned Hospital Readmissions"** answers the second research question by proposing a new deep-learning model called Deep Magnitude Management (D2M) that predicts unplanned hospital readmissions from longitudinal patient data and additional features. In contrast to previously proposed sequential models designed to predict hospital readmission, D2M processes clinical data of successive hospital admissions based on their corresponding recording date and incorporates a mechanism that explicitly focuses on frequent medical events such as chronic diseases. Extensive experiments have demonstrated that D2M improves the accuracy of the unplanned hospital readmissions prediction task.

In summary, the first two publications present new deep-learning models that improve the accuracy of mortality prediction. This improvement is due to the models' ability to overcome underlying problems associated with physiological time series data. In the second paper, the author optimally combines two existing deep learning architectures to propose a model that efficiently encodes underlying medical information, such as diagnostic relationships and visit dates. This combination resulted in improved detection of depression from longitudinal patient claims data. Finally, in the paper under review, the author introduces a novel deep-learning model that processes clinical data of successive admissions while considering their corresponding dates and frequent medical events. Beyond the ability of the introduced models to answer research questions, the author believes that once they reach maturity, they will contribute to the expansion of AI in medicine and, by extension, in other fields. Indeed, many decision support tools in real-world scenarios rely on irregular time series and longitudinal data.

3 Research Methodology

The objective of the thesis is to propose machine learning tools to predict adverse medical events. Therefore, the author uses the Applied Research methodology as a research design since it seeks to provide a practical solution to an existing problem. The existing problem is the prediction of adverse medical events, while the practical solution consists of machine learning tools. Applied Research encompasses systematic procedures involving the identification of problems, and the development of hypotheses, followed by experiments to test those hypotheses. Through these procedures, the author can address the underlying problems related to the prediction of adverse medical events and provide robust artefacts as solutions. Figure 1 illustrates the systematic procedures of Applied Research.

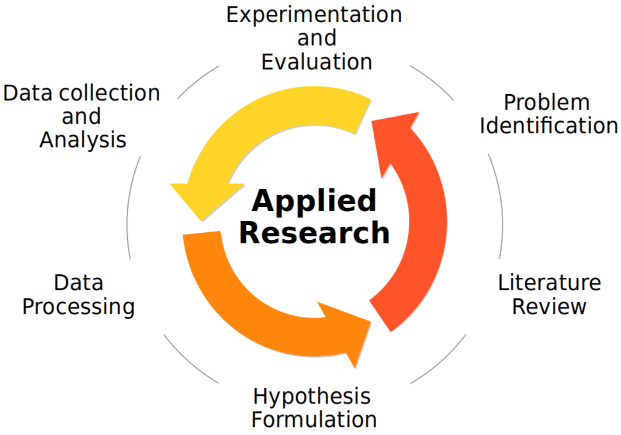


Figure 1: Systematic applied research procedures.

Applied Research methodology has proven to be well-suited to various fields seeking practical solutions to the problems they face [18]. By relying on it, researchers have been able to provide valuable artefacts that contribute to the advancement of their field and, by extension, to other fields. The author therefore considers its adoption as essential to conduct rigorous research that will ultimately provide efficient and robust machine learning tools for predicting adverse medical events and additional artefacts such as peer-reviewed study synthesis and model formulation.

To overcome the thesis challenges, Applied Research iterations were adopted in all three publications and the article under review. These iterations, including evaluating the literature, problem identification, data collection and analysis, data processing, hypotheses formulation, and development of deep learning models implementing these hypotheses, constitute the basic steps in each publication. Publications $[I, III]$ are published in high-impact Q1 journals, highlighting the considerable contribution they make to the application of AI in healthcare. Additionally, Publication $[III]$ is published in a well-regarded B conference. Overall, these publications demonstrate the value and robustness of the work carried out as part of this thesis to provide effective ML tools for predicting adverse events during a patient's medical stay.

To answer the research questions, specific types of Applied Research were adopted. In Publications $[I,II]$ and the article under review, the Research and Development type was used as the backbone. This made it possible to build and evaluate new deep-learning models based on hypotheses formulated from data. In Publication $[III]$, the Evaluation Research type was adopted. After screening the literature, existing deep learning architectures capable of modelling hypotheses formulated from data and medical procedures were adopted. Table 1 presents each research question, the publications that answer it, and the type of applied research adopted in these publications.

Table 1: Research questions, publications, and type of Applied Research.

	Publication	Type of Applied Research
RQ1	I II	**Research and Development**: exploring the literature, formulating hypotheses and building new deep learning models.
RQ2	III	**Evaluation research**: exploring the literature, formulating hypotheses and using existing deep learning architectures to build a model.
RQ2	Under review	**Research and Development**: exploring the literature, formulating hypotheses and building new deep learning models.

In publications $[I,II]$, the Research and Development design is adopted because the author believed that building new deep learning models was the appropriate approach to answer the first research question. Existing machine or deep learning models are suboptimal for predicting mortality from physiological time series. Therefore, it was necessary to analyse their limitations by reviewing the literature, formulating hypotheses, and proposing new models based on these hypotheses. TensorFlow and Keras, two Python machine-learning libraries, were used to develop the models proposed in Publications $[I,II]$. Area Under the ROC Curve (AUC), Area Under the Precision-Recall Curve (AUPRC), and F1 scores, specificity and sensitivity were used as metrics to evaluate the proposed models. These metrics were chosen because the detection and prediction tasks conducted in the thesis involve binary classifications with unbalanced class distributions. They allow for the evaluation of the models' ability to classify the minority classes (deceased patient, depressed patient, patient to be readmitted), which are the classes of interest in each article. The models were tested, evaluated and compared to state-of-the-art models using the publicly available MIMIC-3 [23] and PhysioNet [20] databases. MIMIC-3 consists of anonymized health-related data from over forty thousand patients who stayed in the ICU at Beth Israel Deaconess Medical Center between 2001 and 2012. PhysioNet is a database developed as part of a mortality prediction challenge. It contains twelve thousand patient records of patients hospitalized for cardiac diseases in intensive care units. MIMIC-3 and PhysioNet are benchmark databases widely used in the literature to evaluate medical AI models. Overall, the Research and Development design led to the creation of robust and accurate predictive models that will ultimately improve healthcare.

In publication $[III]$, the author employed the Evaluation Research Design to address the second research question, given that certain pre-existing deep learning architectures already offered the essential framework for incorporating the formulated hypotheses. Two existing architectures were combined to build a model. The latter was coded using the Python programming language and its machine learning libraries, namely TensorFlow and Keras. AUC, AUPRC, sensitivity and specificity were used as metrics. The author trained, tested, and compared the model with various deep learning models considered good candidates for depression detection using Estonian patient medical claims data, which includes more than eighty thousand publicly insured people with a diagnosis of depression. Additionally, Matplotlib, a Python graphics library, was utilized to generate graphs that illustrate the correlation between diseases during the depression detection process.

Finally, in the paper under review, the author once again embarrassed the Research and Development design to develop a new deep-learning model that predicts the risk of patients being readmitted to the hospital. This research design enables the identification of the limitations of existing models designed to predict unplanned hospital readmissions. Based on the findings, the author proposed a new deep-learning model. Python and its machine learning libraries, including TensorFlow and Keras, were used to implement the model. AUC, AUPRC, sensitivity and specificity were used as metrics. The model was trained with data extracted from the MIMIC-3 database. Some graphs were also provided to explain the model's prediction (explainability). They were created using Matplotlib, a Python graphics library.

The Applied Research methodology has made it possible, through meticulous steps, to build robust neural networks to predict adverse medical events effectively.

4 Related Research

This section provides an overview of the literature on predictive models that exploit multivariate clinical time series or longitudinal patient data to predict adverse medical events. A meticulous analysis of existing work is carried out to extract knowledge and identify limitations. The extracted knowledge is synthesized to provide an overview of the literature and construct new models to fill the identified gaps.

4.1 Predictive Models Using Multivariate Clinical Time Series

The development of sensors capable of collecting data continuously over time has made it possible to create powerful predictive machine-learning tools trained from time series data. For instance, in [38] and [55], the authors exploit time series data to propose predictive models for traffic management and stock market prediction, respectively. In the medical field, time series are an effective and rich source of information for analysing a patient's state of health over time. Although medical sensors such as electromyography and pulse oximetry can collect massive amounts of time-series clinical data, these data often need to be analysed and processed before further use for a predictive task. The general problem facing clinical time series is temporal irregularity (i.e. irregular time interval between successive observations), which leads to data sparsity and an increase in missing values. Indeed, time series data may already contain missing values due to outliers or machine failures. Temporal irregularity is due to the divergence in data collection frequency between medical sensors. It leads to irregular univariate time series data when a single sensor is used for time series analysis and to irregular multivariate time series when several sensors are used. Several models have been proposed in the literature to overcome the pitfalls of clinical time series while increasing the accuracy of the downstream task [43, 44, 47]. Although some works have implemented Convolutional Neural Networks (CNNs) and Transformer models to deal with irregular time series [41, 43, 50, 51], RNN-based models are the most preferred.

RNN is the state-of-the-art model for dealing with time series. However, it reaches its limits when these are irregular [37]. Three approaches are often adopted to make it more effective against irregular time series. The first is to discretize the time period over which the data were collected [16, 52]. The problem with this approach is that it requires ad hoc management of time intervals and removes fine-grained information. The second approach is to modify the structure of the RNN so that imputation/interpolation can be performed directly within its core [42, 46]. Although this approach has the advantage of performing data-driven imputation, it often introduces additional noise during the learning process. The third approach is to model the hidden states of RNN as continuous functions via ordinary differential equations (ODE) [25, 26]. Like the second approach, it has the disadvantage of introducing additional noise during the learning process. The noise is due to the fact that ODE-RNN-based models deal with time series in continuous spacetime. Consequently, latent values calculated at irrelevant timestamps will introduce noise. The main idea behind the last two approaches is to process raw irregular time series without any preprocessing step, such as time discretization, that may discard relevant information.

The thesis, aware of these limitations that make medical predictive models suboptimal, introduces new deep-learning architectures through Publications $[I, II]$ dedicated to answering the first research question. An alignment strategy underpins these deep learning architectures, enabling direct processing of raw time-series data while preserving relevant information. Although the model proposed in Publication $[I]$ makes it possible to answer the first research question, it comes up against a limitation, namely the imputation of missing values (caused by outliers or machine failures) during the pre-processing phase. In addition, the imputation methods used are based on strong assumptions. Publication $[II]$ therefore introduces a data-driven imputation and padding approach to fill this gap.

4.2 Predictive Models Using Longitudinal Patient Data

The development of EHRs in medical facilities has undoubtedly contributed to the breakthrough of IA in medicine [3, 57]. EHR makes it possible to save heterogeneous longitudinal patient data. Therefore, decision support tools, such as predictive machine learning models, can leverage historical and current data. It is common for physicians to review patients' historical medical data before any decision-making. Indeed, in some cases, such as that of a patient suffering from a chronic illness, current medical information may not be sufficiently quantitative or qualitative to establish a diagnosis. When the amount of data is not so large, a physician can mine historical medical data for diagnostic purposes. However, this becomes difficult to achieve when historical data has been collected over a long period at each admission or visit.

Establishing a diagnosis often requires browsing through a patient's historical medical data in medical settings. For this reason, machine learning models, in particular deep learning models capable of processing current information while taking into account past information, have been widely used in the literature to address medical problems [14, 28]. As deep learning models such as RNNs and Transformers integrate different strategies to encode long-term relationships between data, they are often preferred as backbones in the literature to tackle medical tasks. Although they have shown functional results, vanilla RNNs and vanilla Transformers treat successive medical events as if they occurred at a regular time interval [21, 27]. This is an incorrect assumption, because the elapsed time between successive medical events may vary. Failing to consider elapsed days in the model's decision process may lead the model to give the same level of importance to past and current data. As a result, the prediction or detection is affected. Another limitation of existing models designed to predict adverse medical events from longitudinal patient data is the lack of a mechanism that explicitly focuses on frequent medical events. Indeed, frequent medical events, such as chronic diseases, are often at the origin of adverse medical events [36]. It is therefore essential to focus explicitly on them in the decision-making process.

Publication $[III]$ combines two existing deep-learning models to detect depression from longitudinal patient data. While one model encodes relationships between diagnoses, the second integrates elapsed days into the decision process. The article, under review, introduces a new deep-learning model that considers elapsed days between admissions and incorporates a mechanism that explicitly focuses on frequent medical events. These different strategies applied in Publication $[III]$ and in the article under review have enabled improving the accuracy of the models dedicated to the detection of depression and the prediction of unplanned hospital readmissions.

5 Publication-specific Contributions

This section presents the deep learning architectures introduced in each publication to answer the research questions. The aim is to give a high-level overview of the components of each architecture and how they work together.

5.1 A New Efficient ALignment-driven Neural Network for Mortality Prediction from Irregular Multivariate Time Series data [I]

In this study, the author introduces a new deep learning architecture called ALignment-driven Neural Network (ALNN) that aims to predict ICU mortality from irregular physiological time series data. The main contribution of ALNN is to overcome the limitations of RNNs, which have difficulty handling irregular time-series data, and thus improve the accuracy of predictions. Its advantages over previous studies presented in Subsection 4.1 include: not performing alignment during the preprocessing step, which could remove relevant information and introduce noise in the model calculations; and filling in only missing values that are caused by issues unrelated to time irregularities such as outliers, sensor failure, or change in the patient's state of health. Indeed, because of changes in the patient's state of health, the physician may decide not to record his/her physiological measurements during a stay or a medical visit.

The proposed overall model, called ALNN-GRU, that was trained and tested with the MIMIC-3 database includes a preprocessing step that involves missing value completion; an alignment component that maps the irregular multivariate time series into a pseudo-regular time series called pseudo-aligned latent values; a GRU that encodes the sequential pattern; and a classifier that predicts whether a patient will die or not. Each stage and component of the model is shown in Figure 2 and detailed in the following paragraphs.

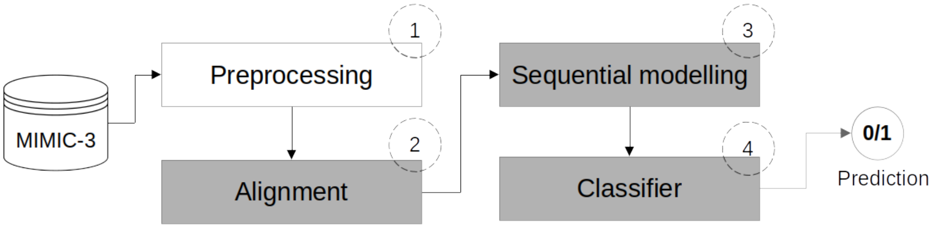


Figure 2: A high-level abstraction of the ALNN-GRU. The white box represents a processing step performed outside the model (before training) and the dark boxes represent steps performed within the model.

- **MIMIC-3** is the database used to train and evaluate the ALNN-GRU. From the chartevents and outputevents tables, we have extracted patient data from the first 24 hours (37,375 patients) and 48 hours (25,755 patients). As a patient may have several admissions, for the 24-hour dataset, we obtained 45,954 admissions records, 41,162 associated with living patients, and 4,792 (11.64%) associated with deceased patients. Whereas, for the 48 hour-dataset, we obtained 30,415 admissions records, 26,577 associated with living patients and 3,838 (12%) associated with deceased patients.
- **Preprocessing** consists of filling in missing values using various techniques, including imputation by empirical mean, interpolation, backward filling and forward filling.

Each of these techniques is used in different contexts. Interpolation is used to fill in missing values between observations; backward filling is employed when initial values are missing; forward filling is utilized to obtain univariate time series with the same number of observations; and the empirical mean is used to complete the variables without observations. The output of the processing step is the imputed irregular multivariate time series;

- **Alignment** aims to transform imputed irregular multivariate time series into pseudo-aligned latent values. This transformation process is carried out by a neural network called ALNN, which first calculates the time lag scores, and then performs value-level extraction and feature-level aggregation. The time lag scores encode the amount of information that must be accounted for in each value based on their temporal distance from each user-defined uniformly spaced reference time point. Once time lag scores are calculated, they are combined individually with their corresponding value, the mask indicator (which indicates whether the corresponding value is observed or imputed) and the time variation value. This step corresponds to value-level extraction. The result of value-level extraction, which is a tensor with each value corresponding to a latent value of each feature at each reference point in time, will be aggregated to produce the pseudo-aligned latent values. This aggregation step is called feature-level aggregation. For example, if the study involves 5 features with each 10 observations and there are 4 reference time points, the tensor obtained at the value-level extraction will be of shape $4 \times 10 \times 5$. In feature-level aggregation, the aggregation will be performed on the second axis of the previously obtained tensor to produce the pseudo-aligned latent value matrix of shape 4×5. The pseudo-aligned latent values matrix is a matrix in which each row corresponds to the latent value of each physiological feature at each evenly spaced reference time point;

- **Sequential modelling** involves encoding the sequential order contained in the pseudo-aligned latent values. This is achieved using a GRU. Initially, the GRU is suboptimal with irregular multivariate time series. Therefore, the pseudo-aligned latent value matrix, which is a pseudo-regular version of the original irregular multivariate time series, makes the GRU more optimal. The GRU output is a vector called context vector. It is a compressed version of the pseud-aligned latent value. It can be seen as a latent vector containing information about the patient's state of health;

- **Classifier** is a set of stacked feedforward neural networks that take the context vector as input and calculate the likelihood of a patient dying. It is worth mentioning that ALNN, GRU and classifier are trained end-to-end for better hyperparameter optimization.

Although the ALNN-GRU model has improved the accuracy of ICU mortality prediction, it still faces a major limitation, namely filling in missing values during preprocessing. The imputation technique used during preprocessing relies on strong assumptions and may introduce noise into the model calculations. A data-driven approach that could be a solution to this limitation is introduced in the next publication.

5.2 Deep Padding and Alignment Strategies for Irregular Multivariate Clinical Time Series [II]

This publication aims to overcome the aforementioned limitations of ALNN-GRU. Instead of imputing missing values during the pre-processing phase, which can be a noise driver, it introduces data-driven imputation and padding strategies, performed via a bidirectional recurrent imputation for time series (BRITS) [10] variant called PaddGRU. BRITS was originally proposed to fill in missing values in time series. However, its architecture does not allow it, under certain conditions, to generate the timestamp values required to run ALNN. It has therefore been redesigned to obtain PaddGRU. PaddGRU performs data-driven imputations to fill in missing values and data-driven padding to obtain univariate time series of equal length and their corresponding timestamp values. The data-driven imputation and padding approaches have the advantage of being guided by the underlying structure of the data and the downstream task criterion. The imputed and padded values obtained using these approaches are more reliable and less noisy, leading to better model performance. Excepted the data-driven imputation and padding component presented in this publication, the other components are those presented in the first publication. A high-level view of the model architecture is shown in Figure 3.

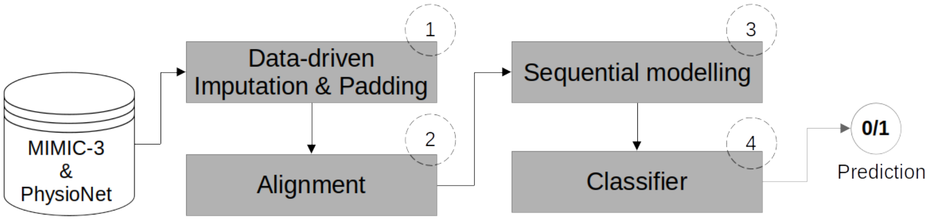


Figure 3: A high-level abstraction of the PaddGRU+ALNN-GRU. The dark boxes represent the steps performed in the model.

- **MIMIC-3** is one of the databases used to train and evaluate the proposed model. In this study, only patients who spend at least 48 are included. $27,162$ patients fulfilling this condition were obtained. As some patients were admitted several times to the ICU, we extracted $32,496$ admissions distributed as follows: $28,075$ related to patients who remain alive and $4,421$ (15.75%) to patients who die. 12 physiological time series data are used as model input.
- **PhysioNet** is the second database used to train and evaluate the proposed model. Data from $4,000$ patients were extracted. They are distributed as follows: $3,446$ related to patients who remain alive and 554 (13.85%) to patients who die. 37 physiological time series data are used as model input.
- **Data-driven imputation and padding** is performed with a PaddGRU that is a BRITS variant. The first step is to generate imputed values via a linear transformation of the hidden layer. The use of hidden layers enables to recursively introduce past information in the generation process. The second step, via another linear transformation of the hidden layer, consists of generating time variation values that will

be added to the previous timestamps to generate the actual ones. Since generated timestamps cannot exceed the highest timestamp value, the previous timestamp is used if a generated timestamp is greater than the highest timestamp value. For instance, if the data was collected over 48 hours, the maximum value of the timestamps will be 48. Therefore, if a generated value is greater than this value, the previously generated timestamp is used instead. This trick helps to preserve the temporal structure of the data. PaddGRU and ALNN-GRU are combined and trained in an end-to-end fashion. Experimental results obtained after training showed that this combination makes the prediction of ICU mortality more accurate.

The contribution made in publications $[I, II]$ is the introduction of a novel deep-learning model that efficiently predicts ICU mortality from irregular multivariate clinical time series. Better prediction of ICU mortality in the short term will help physicians in their decision-making and, in the long term, will improve patients' health and reduce their healthcare costs and those of hospitals.

5.3 Evaluation of Deep Learning-Based Depression Detection Using Medical Claims Data [III]

In real-world case scenarios, practitioners are used to analysing the diagnoses made during the patient's most recent admissions or visits. This gives them a clearer picture of the patient's health trajectory and enables them to make an accurate diagnosis. While this process can be achieved with a relatively small amount of data, it is almost impossible, even for an experienced practitioner, when faced with a large amount of data with complex relationships. This publication therefore aims to combine two existing deep learning architectures, which enable accurate decision-making from large amounts of longitudinal patient data while efficiently encoding the relationship and temporal information of clinical data considered for decision-making. The medical event studied in this publication is the detection of depression from longitudinal patients' claims data during their visit (or medical stay).

The proposed model, called Att-GRU-decay, comprises two neural network layers, Self-Attention and GRU-decay [12]. The Self-Attention layer, a sublayer of the vanilla Transformer [49], is used to encode the diagnoses (represented in ICD-10 format) based on their hidden relationships. On the other hand, GRU-decay is responsible for weighting the significance of diagnoses according to their recording date and modelling their sequential order. On top of Self-Attention and GRU-decay, there is an embedding layer that aims to encode each diagnosis into a continuous vector before any further use. A high-level abstraction of Att-GRU-decay is illustrated in Figure 4. Each component is described in the following paragraphs.

- The **Medical Claims data** used consists of all publicly insured people in Estonia with a depression diagnosis ($80,243$ patients with $4,252,213$ diagnoses). The control group consists of $732,610$ patients (with $22,721,730$ diagnoses), of which $498,764$ people (with $10,779,835$ diagnoses) did not have a psychiatric disorder diagnosed and $233,846$ patients (with $11,941,895$ diagnoses) had a psychiatric disorder other than depression.

- **Diagnosis embedding** involves encoding each diagnosis in a continuous vector via an embedding layer. These continuous vectors are called diagnoses embeddings;

- **Encoding of relationship** aims to learn and encode the relationship between relevant diagnosis embeddings and filter out irrelevant diagnosis embeddings to the

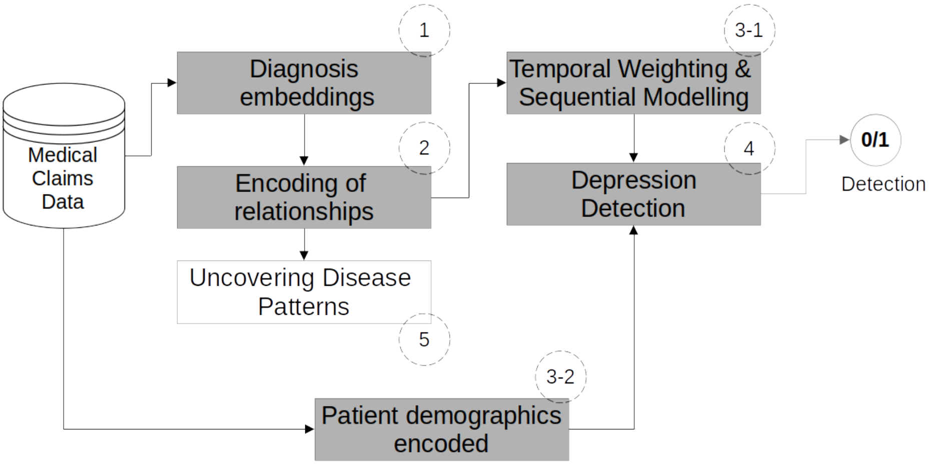

1
3-1
Diagnosis embeddings
Temporal Weighting & Sequential Modelling
Medical Claims Data
2
4
0/1
Detection
Encoding of relationships
Depression Detection
Uncovering Disease Patterns
5
3-2
Patient demographics encoded

Figure 4: A high-level abstraction of Att-GRU-decay.The white box represents a step performed outside the model (after training) and the dark boxes represent steps performed within the model.

downstream task. This step is carried out in the Self-Attention layer;

- **The temporal weighting and sequential modelling** performed in GRU-decay, consists of weighting the significance of each diagnosis (embedding version) according to their corresponding elapsed days value and modelling the sequential order in which diagnoses were recorded. The elapsed-day value of a diagnosis is obtained by subtracting its corresponding recorded date from that of the following diagnosis. This is linearly transformed and passed to an exponential decay function to produce the decay factor. The decay factor, which is used as a weight, is then multiplied by the actual hidden state. The underlying intuition is to reduce the values of the hidden state when the diagnosis was made a long time ago, and to keep them virtually unchanged when the diagnosis was made recently. The GRU-decay output is a context vector, which is a compressed representation of all diagnosis embeddings and their associated information, including their recording date and sequential order;

- **Patient demographics encoded** is carried out with a feedforward neural network. It aims to encode the patient's demographic data, including gender and age, into a continuous vector. This continuous vector is the latent representation of the patient's demographic;

- **Depression detection** is performed using a set of stacked feed-forward neural networks that calculate the likelihood that a patient is detected as depressed. These feedforward neural networks are fed by the concatenation of the context vector and the latent representation of the patient's demographics.

Although the proposed Att-GRU-decay model is highly accurate in detecting in advance patients who may suffer from depression, its accuracy does not guarantee its use in real-world medical settings. Indeed, in a field as sensitive as medicine, the model's decision must be explained. The author therefore suggested providing physicians with graphs **uncovering disease patterns**. More specifically, these graphs show the correlation between diagnoses so that physicians can better understand what led to the decision.

The study carried out in this publication provides an answer to the second research question. It reveals that better encoding of relationships between clinical codes and considering the elapsed time between visits are crucial factors when predicting or detecting certain medical events from longitudinal patient data spanning several successive visits.

5.4 Deep Magnitude Management of Clinical Code Embeddings to Predict Unplanned Hospital Readmissions [under review]

This paper presents a novel sequential deep-learning architecture, which aims to provide an alternative solution to the second research question. It presents a deep learning model called Deep Magnitude Management (D2M) that predicts unplanned hospital readmissions from longitudinal data of heterogeneous patients. Heterogeneous patients refer to patients with different diseases. Like the model proposed in Publication $[III]$, D2M also integrates into the model calculation scheme the elapsed days between successive admissions. Additionally, it incorporates a mechanism that enables it to focus explicitly on frequent medical events such as chronic diseases, which are often the origin of adverse medical events. Extensive experiments conducted with data extracted from the MIMIC-3 database demonstrated that these different strategies, namely taking into account the days elapsed between admissions and incorporating a mechanism explicitly focused on frequent medical events, improve prediction. A high-level abstraction of D2M is illustrated in Figure 5. Each component is described in the following paragraphs.

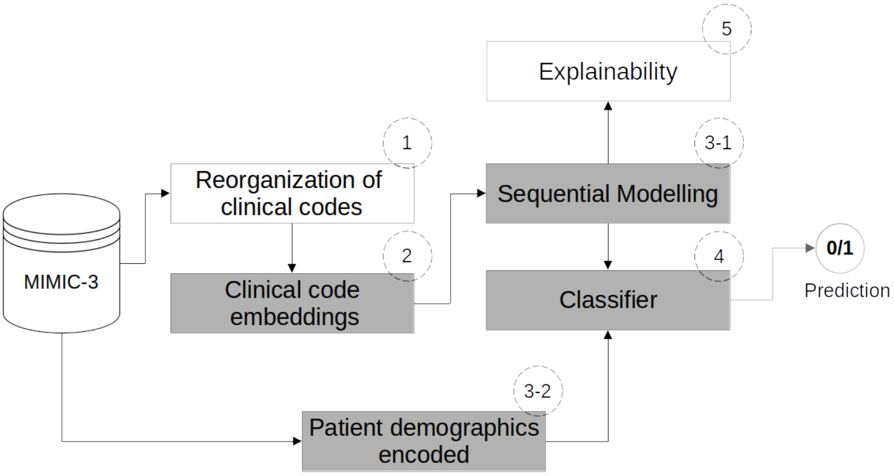


Figure 5: A high-level abstraction of D2M. The white boxes represent steps performed outside the model (before and after training) and the dark boxes represent steps performed within the model.

- The **MIMIC-3** database is used to train and evaluate the D2M model. Data from $14,753$ patients were extracted. $2,471$ (16.75%) are linked to unplanned hospital readmissions (positive cases) and $12,282$ are not (negative cases).

- **Reorganization of clinical codes** rearranges the clinical code in each admission representation. Admissions are represented in vector format. Each value in these vectors is an integer value corresponding to a clinical code. If a clinical code is present in two successive admissions, the rearrangement procedure will place it in the same position index in both admission representations;

- **Clinical code embeddings** maps clinical codes initially represented by integers into continuous vectors. These vectors are called clinical code embeddings;
- **Sequential Modelling** models the sequential order of admissions while transferring information from one admission to another. The information transfer between successive admissions only takes place if and only if there is at least one clinical code embedding belonging to both admissions. The information transfer score, which determines the amount of information to be transferred, is obtained by a nonlinear transformation of the elapsed days between admissions and the similar score of these admissions. This step is performed with a neural network layer called magnitude management;
- **Patient demographics encoded** encodes patient demographics into a continuous vector using a feedforward neural network;
- **Classifier** determines whether a patient will be readmitted to the hospital based on encoded patient demographics and the output of magnitude management, which is the latent representation of admissions. The classifier consists of a set of feedforward neural networks;
- **Explainability** is provided through a set of graphs. These graphs present the diagnosis embeddings in a two-dimensional space, along with their corresponding decay factor and transfer score (if computable). These are used to quantify the contribution of clinical codes in the prediction.

The strategies implemented in D2M improved the prediction of unplanned hospital readmissions, thus answering the second research question. This highlights their importance when processing longitudinal patient data recorded over multiple admissions. D2M performances must be improved before any deployment in a real-world medical scenario. In future work, the use of additional features, such as laboratory results and physiological measurements, will be investigated to improve its performance.

6 Discussion of Challenges

New deep-learning architectures presented in the publications have proven effective in predicting adverse medical events during patient stays. While Publication $[II]$ introduced various strategies to overcome a limitation of the model presented in Publication $[I]$, the model developed in Publication $[II]$ still faces major limitations. Some limitations are also identified in Publication $[III]$ and in the article under review. Aware of the domain's sensitivity and the impact that a prediction error can have on patient health, future work will focus on overcoming these limitations before deploying the proposed models in real-world medical scenarios.

Although the model developed in Publication $[II]$ overcomes a limitation of the one proposed in Publication $[I]$, it still faces a major limitation, namely lack of explainability. In a field as sensitive as medicine, it is not just a matter of providing accurate models. It is equally important that these models are explainable. Explainability is a must in medicine [22]. If this is very important from a social or legal perspective, it enables healthcare professionals in general and physicians in particular to understand what has led to the decision. How the model makes the decision must be transparent and explainable. Several studies have shown that deep models struggle to be adopted in real-world medical scenarios due to their lack of explainability [1, 15]. It is therefore crucial to make them explainable to increase the level of confidence of physicians. Various fields, including medicine, have widely used the attention mechanism, a procedure that involves weighting the most relevant model inputs, to make model predictions interpretable and explainable. More recently, Self-Attention introduced in the Vanilla Transformer has become the state-of-the-art technique adopted to provide explainable deep learning models. Although these techniques are potential solutions to make the proposed model explainable, an innovative technique would be the analysis of the activation weights associated with the pseudo-aligned latent values. Indeed, we can exploit the pseudo-aligned latent values to easily identify the latent value associated with each feature at different timestamps. Consequently, analysing the weight associated with these latent values can help identify which feature and at which period contributes most to the prediction.

Taking into account the degree of uncertainty in the imputed value is also an aspect on which future work will focus. Not all imputed values have the same level of uncertainty. While those with a low level of uncertainty will help to improve the prediction, those with a high level of uncertainty will tend to reduce the model's performance. One possible solution to fill this gap is to use probabilistic techniques such as those used in studies [24, 33], to obtain the level of uncertainty in the imputed value and inject it into the decision-making process.

Another limitation of the models introduced in Publications $[I]$ and $[II]$ is the exclusive use of physiological measurements as predictors. Although these are the most reliable initial data that physicians can obtain in the ICU, additional predictors such as demographics, prior medical events, images, and others can be incorporated into the decision-making process. The author then intends to expand the model architecture to process and fuse additional predictors with the pseudo-aligned latent values, thereby improving the prediction of ICU mortality.

Regarding publication $[III]$, the study may be subject to bias. Medical claims data are used for billing purposes. There is no guarantee that some diagnoses made by physicians are not made solely to increase their revenue. In addition, because of the previously described low accuracy of human diagnoses, the ground truth used to train the model likely introduced biases into the prediction. This aspect of data quality needs to be further investigated. The model should also be evaluated on non-medical claims data to further support the research findings. Although the model has performed spectacularly well using patients' diagnostic and demographic data, the author plans to see what benefits can be gained by introducing interview notes into the decision-making process. These could be used to improve the explanatory power of the model by identifying the most relevant clinical terms in the notes.

Finally, for the model proposed in the article under review, the author intends to train it with more clinical features and improve its explanatory power. Since some clinical features might be represented in time series format or continuous values, the author plans to combine the proposed model (D2M) in an end-to-end fashion with other deep learning architectures such as CNN, Transformers or/and RNN that will be dedicated to processing these additional features.

In addition to the limitations listed, the author plans to carefully study and consider the ethical aspects of using the data and models. When developing and deploying AI models in medical settings, these must be accompanied by frameworks determining responsibilities and legal liability, fair use of data must be ensured, and transparent reporting on model performance must be available [40]. For example, suppose the model predicts with a 98% probability that the patient will die. Several questions arise: Should the doctor stop or continue monitoring? Should the doctor reduce resources in favour of patients with a low risk of death? These are just a few examples of the ethical issues that need to be carefully analysed before any further deployment.

In summary, while the proposed models have met the challenges of predicting medical events encountered in the literature, they also have certain limitations. These limitations include lack of explainability, inability to account for uncertainty in the imputed value, and the limited number of clinical features used as predictors. To address these limitations and make AI models in medicine trustworthy, future work will be directed towards developing an explainable component to identify the features in the clinical time series that have contributed to the decision; the integration of uncertainty into the model calculation scheme so that the model can rely more on low-uncertainty imputed values, thereby reducing the noise that high-uncertainty values can introduce; and using additional clinical features as predictors to improve model performance. Ethical issues and biases in models and data will also be addressed.

7 Conclusion

This thesis presents novel deep-learning models designed to predict adverse medical events from multivariate clinical time series and longitudinal patient data. Prediction of ICU mortality, detection of depression and prediction of unplanned hospital readmission are the medical tasks that the introduced models seek to improve. While multivariate clinical time series are used to predict ICU mortality, longitudinal patient data are used to detect depression at an early stage and predict unplanned hospital readmissions. Using data extracted from MIMIC-3, PhysioNet and those provided by the Estonian Health Insurance Fund (medical claims data), extensive experiments were conducted to evaluate and compare the proposed models to state-of-the-art models. The superior performance of the model dedicated to ICU mortality prediction revealed the crucial importance of data-driven alignment, imputation and padding when predictions are based on irregular multivariate clinical time series. For the depression detection task, better encoding of relationships between diagnoses and incorporation of elapsed days between visits into the model calculation scheme proved to be relevant factors when using longitudinal patient data as predictors. Finally, for the unplanned hospital readmissions prediction task, the experimental results highlighted the importance of taking into account the days elapsed between admissions, and the importance of implementing a mechanism that focuses on frequent medical events. Overall, the various experiments carried out as part of the thesis revealed the crucial importance of temporal information in predictive medical tasks. This thesis proposes not only efficient models for irregular numerical temporal data but also efficient models for irregular categorical temporal data.

The thesis has a short-term and long-term goal. On the one hand, the short-term goal is to propose machine learning tools, more precisely deep learning models that effectively predict adverse medical events. On the other hand, its long-term goal is to make the proposed models explainable more accurate and robust so that they can be adopted in real medical scenarios to support physicians in their daily decision-making. The author is convinced that the proposed models will make it possible to improve the delivery of care, better manage resources, relieve hospital overcrowding, avoid physician burnout, improve patient health and reduce healthcare costs for patients and hospitals. It should be noted that the machine learning tools developed in this thesis are not intended to replace healthcare professionals, but rather to assist them.

To achieve the thesis's long-term goal, the limitations of the proposed models must first be overcome. For the ICU mortality prediction model, future work will account for analysing the uncertainty of imputed values, explaining the model prediction, and training it with additional clinical features and patient demographics. For the study on depression detection and prediction of unplanned hospital readmissions, future work will focus on improving the explanatory power of their respective model and training these models with additional clinical features. In addition to limitations, ethical questions and biases in models and data will also be addressed. Once these limitations are overcome and ethical and bias issues addressed, the author plans to integrate all models into a unified system. This will enable each model to benefit not only from the hidden patterns extracted by other models in the data but also from the predictions of these models. In parallel, the models will be evaluated on additional medical and nonmedical tasks to determine whether they can be generalized.

List of Figures

1 Systematic applied research procedures 16
2 A high-level abstraction of the ALNN-GRU 21
3 A high-level abstraction of the PaddGRU+ALNN-GRU 23
4 A high-level abstraction of Att-GRU-decay 25
5 A high-level abstraction of D2M 26

List of Tables

1 Research questions, publications, and type of Applied Research. 17

References

[1] A. O. Akinrinmade, T. M. Adebile, C. Ezuma-Ebong, K. Bolaji, A. Ajufo, A. O. Adigun, M. Mohammad, J. C. Dike, and O. E. Okobi. Artificial intelligence in healthcare: Perception and reality. *Cureus*, 15(9), 2023.

[2] A. Ashfaq, A. Sant'Anna, M. Lingman, and S. Nowaczyk. Readmission prediction using deep learning on electronic health records. *Journal of biomedical informatics*, 97:103256, 2019.

[3] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 43–51, 2018.

[4] E. Beede, E. Baylor, F. Hersch, A. Iurchenko, L. Wilcox, P. Ruamviboonsuk, and L. M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–12, 2020.

[5] M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, and D. Draheim. Evaluation of deep learning-based depression detection using medical claims data. *Artificial Intelligence in Medicine*, 147:102745, 2024.

[6] N. Bignoumba and S. Ben Yahiaand N. Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. In *Proceedings of KES'2024 - the 28th Annual KES Conference*, 2024.

[7] N. Bignoumba, M. Bertl, S. B. Yahia, and N. Mellouli. Deep magnitude management of clinical code embeddings to predict unplanned hospital readmissions. *PREPRINT (Version 2) available at Research Square*, 2024.

[8] N. Bignoumba, N. Mellouli, and S. B. Yahia. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. *Expert Systems with Applications*, 238:122148, 2024.

[9] N. Bignoumba, S. B. Yahia, and N. Mellouli. Étude de similarité des patients pour identifier les unités hospitalières ayant le taux le plus élevé de réadmissions non planifiées. *Actes de la journée d'étude sur la Similarité entre Patients, SimPa 2023*, page 24, 2023.

[10] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[11] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. Prudhvith, and N. Kumar. Image classification using svm and cnn. In *2020 International conference on computer science, engineering and applications (ICCSEA)*, pages 1–5. IEEE, 2020.

[12] 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):1–12, 2018.

[13] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[14] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

[15] T. Davenport and R. Kalakota. The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94, 2019.

[16] N. El-Rashidy, S. El-Sappagh, T. Abuhmed, S. Abdelrazek, and H. M. El-Bakry. Intensive care unit mortality prediction: An improved patient-specific stacking ensemble model. *IEEE Access*, 8:133541–133564, 2020.

[17] S. Elloumi and N. Bignoumba. Stacking deep-learning model, stories and drawing properties for automatic scene generation. *International Journal of Advanced Computer Science and Applications*, 14(1), 2023.

[18] A. Everitt, P. Hardiker, J. Littlewood, and A. Mullender. *Applied research for better practice*. Bloomsbury Publishing, 1992.

[19] A. L. Fradkov. Early history of machine learning. *IFAC-PapersOnLine*, 53(2):1385–1390, 2020.

[20] 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):e215–e220, 2000.

[21] M. Gupta, T.-L. T. Phan, H. T. Bunnell, and R. Beheshti. Obesity prediction with ehr data: A deep learning approach with interpretable elements. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(3):1–19, 2022.

[22] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1312, 2019.

[23] A. E. 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(1):1–9, 2016.

[24] E. Jun, A. W. Mulyadi, J. Choi, and H.-I. Suk. Uncertainty-gated stochastic sequential model for ehr mortality prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4052–4062, 2020.

[25] M. Lechner and R. M. Hasani. Learning long-term dependencies in irregularly-sampled time series. *CoRR*, abs/2006.04418, 2020.

[26] L. Li, J. Yan, Y. Zhang, J. Zhang, J. Bao, Y. Jin, and X. Yang. Learning generative rnn-ode for collaborative time-series and event sequence forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7118–7137, 2022.

[27] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020.

[28] X. Luo, P. Gandhi, Z. Zhang, W. Shao, Z. Han, V. Chandrasekaran, V. Turzhitsky, V. Bali, A. R. Roberts, M. Metzger, et al. Applying interpretable deep learning models to identify chronic cough patients using ehr data. *Computer Methods and Programs in Biomedicine*, 210:106395, 2021.

[29] K.-K. Mak, Y.-H. Wong, and M. R. Pichika. Artificial intelligence in drug discovery and development. *Drug Discovery and Evaluation: Safety and Pharmacokinetic Assays*, pages 1–38, 2023.

[30] S. Maurya, S. Tiwari, M. C. Mothukuri, C. M. Tangeda, R. N. S. Nandigam, and D. C. Addagiri. A review on recent developments in cancer detection using machine learning and deep learning models. *Biomedical Signal Processing and Control*, 80:104398, 2023.

[31] C. K. McIlvennan, Z. J. Eapen, and L. A. Allen. Hospital readmissions reduction program. *Circulation*, 131(20):1796–1803, 2015.

[32] L. N. Mintarya, J. N. Halim, C. Angie, S. Achmad, and A. Kurniawan. Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Computer Science*, 216:96–102, 2023.

[33] A. W. Mulyadi, E. Jun, and H.-I. Suk. Uncertainty-aware variational-recurrent imputation network for clinical time series. *IEEE Transactions on Cybernetics*, 52(9):9684–9694, 2021.

[34] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. S. Deep learning for stock market prediction. *Entropy*, 22(8):840, 2020.

[35] A. Naemi, T. Schmidt, M. Mansourvar, M. Naghavi-Behzad, A. Ebrahimi, and U. K. Wiil. Machine learning techniques for mortality prediction in emergency departments: a systematic review. *BMJ open*, 11(11):e052663, 2021.

[36] M. D. Naylor, D. Brooten, R. Campbell, B. S. Jacobsen, M. D. Mezey, M. V. Pauly, and J. S. Schwartz. Comprehensive discharge planning and home follow-up of hospitalized elders: a randomized clinical trial. *Jama*, 281(7):613–620, 1999.

[37] D. Neil, M. Pfeiffer, and S.-C. Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. *Advances in neural information processing systems*, 29, 2016.

[38] C. Ounoughi and S. B. Yahia. Sequence to sequence hybrid bi-lstm model for traffic speed prediction. *Expert Systems with Applications*, 236:121325, 2024.

[39] R. Patel, A. E. Arisoyin, O. U. Okoronkwo, S. Aruoture, O. E. Okobi, M. Nwankwo, E. Okobi, F. Okobi, and O. E. Momodu. Trends and factors associated with the mortality rate of depressive episodes: an analysis of the cdc wide-ranging online data for epidemiological research (wonder) database. *Cureus*, 15(7), 2023.

[40] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

[41] H. Shi, Y. Zhang, H. Wu, S. Chang, K. Qian, M. Hasegawa-Johnson, and J. Zhao. Continuous cnn for nonuniform time series. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3550–3554. IEEE, 2021.

[42] 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.

[43] S. N. Shukla and B. M. Marlin. Multi-time attention networks for irregularly sampled time series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[44] S. N. Shukla and B. M. Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[45] N. Sigger, Q.-T. Vien, S. V. Nguyen, G. Tozzi, and T. T. Nguyen. Unveiling the potential of diffusion model-based framework with transformer for hyperspectral image classification. *Scientific Reports*, 14(1):8438, 2024.

[46] Q. Suo, W. Zhong, G. Xun, J. Sun, C. Chen, and A. Zhang. Glima: Global and local time series imputation with multi-directional attention learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 798–807. IEEE, 2020.

[47] S. Tipirneni and C. K. Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

[48] O. Tonkikh, E. Shadmi, N. Flaks-Manov, M. Hoshen, R. D. Balicer, and A. Zisberg. Functional status before and during acute hospitalization and readmission risk identification. *Journal of hospital medicine*, 11(9):636–641, 2016.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] F. Viton, M. Elbattah, J.-L. Guérin, and G. Dequen. Heatmaps for visual explainability of cnn-based predictions for multivariate time series with application to healthcare. In *2020 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–8. IEEE, 2020.

[51] Z. Wang, Y. Zhang, A. Jiang, J. Zhang, Z. Li, J. Gao, K. Li, C. Lu, and Z. Ren. Improving irregularly sampled time series learning with time-aware dual-attention memory-augmented networks. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3523–3527, 2021.

[52] T. Wanyan, H. Honarvar, A. Azad, Y. Ding, and B. S. Glicksberg. Deep learning with heterogeneous graph embeddings for mortality prediction from electronic health records. *Data Intelligence*, 3(3):329–339, 2021.

[53] R. M. Wolf, R. Channa, M. D. Abramoff, and H. P. Lehmann. Cost-effectiveness of autonomous point-of-care diabetic retinopathy screening for pediatric patients with diabetes. *JAMA ophthalmology*, 138(10):1063–1069, 2020.

[54] C. Xiao, T. Ma, A. B. Dieng, D. M. Blei, and F. Wang. Readmission prediction via deep contextual embedding of clinical concepts. *PloS one*, 13(4):e0195024, 2018.

[55] A. Yadav, C. Jha, and A. Sharan. Optimizing lstm for time series prediction in indian stock market. *Procedia Computer Science*, 167:2091–2100, 2020.

[56] W. Yuan, G. Neubig, and P. Liu. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277, 2021.

[57] J. Zhang, K. Kowsari, J. H. Harrison, J. M. Lobo, and L. E. Barnes. Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record. *IEEE Access*, 6:65333–65346, 2018.

Acknowledgements

I would particularly like to thank my thesis supervisors, Professor Sadok Ben Yahia and Professor Nedra Mellouli, for giving me the opportunity to acquire new knowledge through this thesis subject that they entrusted to me. Throughout this long and instructive journey, they taught me how to become a good and resilient researcher. Their corrections, lessons, comments, feedback, and advice were a great help in organizing my research and writing the articles and my manuscript. Once again, thank you!

I would also like to express my sincere appreciation to my colleagues, with whom I have collaborated and had fruitful exchanges. I want to thank them all for the help they have given me throughout my thesis journey. Despite their busy schedules and deadlines, they never hesitated to support me when I needed help. I would like to extend my warmest thanks to the administrative staff, who spared no effort to help me or provide me with the information I needed, whether on the university or other matters.

To all my family and friends in Estonia and around the world who have supported me directly or indirectly, from far and near, I say thank you. Finally, I would like to express my deepest thanks to my father and mother, who never stopped believing in me, even at the worst moments of my school career. Dear parents, thank you for teaching me to be brave. I am sure that I will never be able to return all the material and immaterial goods that you have given me. However, I will continue to move forward and in the right direction to always make you proud.

Abstract

Predictive Systems Using Machine Learning Tools to Forecast Adverse Events During Medical Stays

This Ph.D. thesis uses deep learning, a subfield of machine learning, to propose robust and accurate models for predicting adverse medical events during a patient's medical stay. Indeed, the architecture of deep learning models makes it possible to extract relevant hidden features from massive amounts of data and to process sequential data while preserving relevant past information over time. These characteristics make them ideal candidates for processing heterogeneous and complex medical data, often recorded over successive admissions or visits.

AI tools, such as machine learning and deep learning, have demonstrated significant effectiveness in solving problems across various fields, including finance, engineering, meteorology, and medicine, the main focus of this thesis. Despite the reservations and justified fears that AI in medicine faces compared to other fields, research aimed at proposing AI solutions to medical problems is rapidly expanding. Continuing the trend of previous studies, this thesis focuses on the advancement of AI solutions for medical challenges. More specifically, it aims to solve the problem of predicting adverse medical events during a patient's medical stay. The author adopted an applied research methodology to achieve this objective. Applied research involves effective procedures for identifying the research problem, developing hypotheses, and proposing practical solutions based on these hypotheses.

The machine learning tools developed during the thesis, namely deep learning models, are used to predict mortality, detect depression and predict unplanned hospital readmissions. While physiological time series are used for mortality prediction, longitudinal patient data and additional features are used to detect depression and predict unplanned hospital readmission. Using Applied Research Methodology, the author highlighted the challenges of processing physiological time series and longitudinal patient data, as well as the limitations of state-of-the-art models.

The literature has proposed several models to overcome the challenges associated with irregular clinical time series. Irregular clinical time series can lead to underlying problems such as sparsity, increased missing values, and data misalignment. While some advanced models can produce functional results, they still face certain limitations, including the incorporation of techniques that introduce noise into the model's computational scheme. To address this problem, this thesis presents a novel deep-learning architecture that enables data-driven imputation, padding, and alignment. In extensive experiments, these strategies have proven to be more efficient and generate less noise than those implemented by competing models, thereby increasing the accuracy of mortality prediction.

Two problems have been identified in state-of-the-art models developed to predict adverse medical events from longitudinal patient medical data, i.e. data collected across successive admissions or visits. These include the failure to take into account elapsed days between admission (or visits) and the absence of components focusing explicitly on frequent medical events such as chronic diseases, which are commonly responsible for adverse events. If the model disregards elapsed days, it might assign equal importance to medical events that happened long ago and those that occurred recently. This does not correspond to reality, because in most cases, physicians will base their diagnoses on recent events. Since failing to consider the aforementioned aspect makes previously proposed models suboptimal, this thesis introduces novel deep-learning models that integrate them into the decision-making process to detect depression and predict unplanned hospital

readmission effectively.

This thesis presents the potential of machine learning tools for predicting adverse medical events. It identifies relevant aspects of the data that should be considered when processing health data and proposes various new deep-learning models that incorporate them. Overall, it introduces diverse effective models for processing irregular temporal numerical data and irregular temporal categorical data. The thesis's short-term goal involves proposing effective models to predict adverse medical events. However, its long-term objective involves maturing these models by overcoming their limitations and enabling their adoption in real-world medical scenarios to enhance healthcare delivery.

Kokkuvõte
Ennustavad süsteemid, mis kasutavad masinõppe vahendeid kõrvalekallete prognoosimiseks haiglas viibimise ajal

See doktoritöö kasutab süvaõpet, masinõppe alaharu, et pakkuda välja usaldusväärseid ja täpseid mudeleid meditsiiniliste tüsistuste prognoosimiseks patsiendi hospitaliseerimise ajal. Süvaõppe mudelite arhitektuur võimaldab relevantsete varjatud tunnuste eraldamist massiivsetest andmetest ning järjestikuste andmete töötlemist, säilitades aja jooksul olulist varasemat teavet. Need omadused teevad süvaõppe mudelitest ideaalsed kandidaadid keeruliste ja heterogeensete meditsiiniliste andmete töötlemiseks, mis on sageli salvestatud järjestikuste hospitaliseerimiste või visiitide käigus.

Tehisintellekti tööriistad, nagu masinõpe ja süvaõpe, on näidanud märkimisväärset tõhusust probleemide lahendamisel mitmes valdkonnas, sealhulgas rahanduses, inseneriteaduses, meteoroloogias ja meditsiinis, mis on käesoleva töö põhivaldkond. Vaatamata meditsiinis kasutatava tehisintellekti suhtes valitsevatele kahtlustele ja õigustatud hirmudele võrreldes teiste valdkondadega, on teadustöö, mille eesmärk on pakkuda meditsiinilistele probleemidele tehisintellekti abil lahendusi, kiiresti laienemas. Jätkates varasemate uuringute suundumust, keskendub käesolev töö tehisintellekti lahenduste arendamisele meditsiiniliste väljakutsete lahendamiseks. Täpsemalt püüab see lahendada probleemi, mis seisneb meditsiiniliste tüsistuste ennustamises patsiendi hospitaliseerimise ajal. Selle eesmärgi saavutamiseks kasutas autor rakendusuuringute metoodikat. Rakendusuuringud hõlmavad tõhusaid protseduure uurimisprobleemi tuvastamiseks, hüpoteeside väljatöötamiseks ja nendele hüpoteesidele tuginevate praktiliste lahenduste pakkumiseks.

Doktoritöö käigus välja töötatud masinõppe tööriistu, eelkõige süvaõppe mudeleid, kasutatakse suremuse ennustamiseks, depressiooni tuvastamiseks ja planeerimata hospitaliseerimiste ennustamiseks. Suremuse ennustamiseks kasutatakse füsioloogilisi aegridasid; depressiooni tuvastamiseks ja planeerimata hospitaliseerimise ennustamiseks aga longituudseid andmed ja muid lisatunnuseid. Rakendusuuringute metoodikat kasutades tõi autor esile füsioloogiliste aegridade ja longituudsete andmete töötlemisega seotud probleemid ning ka hetkel kasutatavate mudelite piirangud.

Teaduskirjanduses on välja pakutud mitmeid mudeleid, et ületada ebaregulaarsete kliiniliste aegridadega seotud probleeme. Ebaregulaarsed kliinilised aegread võivad põhjustada selliseid probleeme nagu sisuvaesus, suurenenud puuduvate väärtuste hulk ja andmete joondamise probleemid. Kuigi mõned täiustatud mudelid suudavad anda funktsionaalseid tulemusi, seisavad need siiski silmitsi teatud piirangutega, sealhulgas lisanduv müra mudeli arvutuslikku skeemi. Selle probleemi lahendamiseks pakub käesolev töö välja uue süvaõppe arhitektuuri, mis võimaldab andmepõhist imputeerimist, täidistamist ja joondamist. Ulatuslikes katsetes on need strateegiad osutunud tõhusamaks ja tekitanud vähem müra kui konkureerivate mudelite rakendatud meetodid, suurendades seeläbi suremuse ennustamise täpsust.

Kaks peamist probleemi on tuvastatud tippmudelites, mis on välja töötatud meditsiiniliste tüsistuste ennustamiseks pikisuunalistest patsiendiandmetest, st andmetest, mis on kogutud järjestikuste hospitaliseerimiste või visiitide käigus. Need hõlmavad päevade möödumise mittearvestamist hospitaliseerimiste (või visiitide) vahel ja komponentide puudumist, mis keskenduksid selgesõnaliselt sagedastele meditsiinilistele sündmustele, nagu kroonilised haigused, mis on sageli tüsistuste põhjustajaks. Kui mudel ei arvesta möödunud päevi, võib see omistada sama tähtsust ammu toimunud ja hiljuti aset leidnud meditsiinilistele sündmustele. See ei vasta tegelikkusele, sest enamikul juhtudel tuginevad arstid oma diagnoosides hiljutistele juhtumitele. Kuna eelnimetatud aspektide mittearvestamine muudab varem pakutud mudelid vähem optimaalseks, tutvustab käesolev töö uusi süvaõppe mudeleid, mis integreerivad need otsustusprotsessi, et tõhusalt tuvastada depressiooni ja ennustada planeerimata hospitaliseerimist.

See doktoritöö tutvustab masinõppevahendite potentsiaali meditsiiniliste tüsistuste ennustamiseks. See rõhutab olulisi andmete aspekte, mida tuleks terviseandmete töötlemisel arvesse võtta, ja pakub välja mitmeid uusi süvaõppe mudeleid, mis neid aspekte arvestavad. Kokkuvõttes tutvustab töö erinevaid tõhusaid mudeleid ebaregulaarsete ajaliselt muutuvate numbriliste andmete ja ebaregulaarsete ajaliselt muutuvate kategooriliste andmete töötlemiseks. Doktoritöö lähem eesmärk on pakkuda tõhusaid mudeleid meditsiiniliste tüsistuste ennustamiseks, pikaajaline eesmärk on mudelite täiustamine, nende piirangute ületamine ja rakendamise võimaldamine reaalses meditsiinilises keskkonnas, et parandada tervishoiuteenuste osutamist.

Appendix 1

I

N. Bignoumba, N. Mellouli, and S. B. Yahia. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. *Expert Systems with Applications*, 238:122148, 2024

Expert Systems With Applications 238 (2024) 122148

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new efficient ALignment-driven Neural Network for Mortality Prediction from Irregular Multivariate Time Series data

Nzamba Bignoumba [a,*], Nedra Mellouli [b], Sadok Ben Yahia [a,c]

[a] *Tallinn University of Technology, Akadeemia tee 15a Tallinn 12618, Estonia*
[b] *Léonard de Vinci Pôle Universitaire, Research Center, 92 916 Paris La Défense & LIASD-Université Paris 8 Paris, France*
[c] *University of Southern Denmark, Alsion 2S, Sønderborg 6400, Denmark*

ARTICLE INFO

Keywords:
Aligned data
Deep learning
Electronic Health Records
Irregular multivariate time series

ABSTRACT

The irregularity of the time interval between observations in and across the stream is a key factor that leads to a drop in performance when classical machine learning or deep learning models are used for a downstream task requiring multivariate time series. Indeed, irregular multivariate time series not only increase the rate of missing values but also lead to data sparsity, which consequently makes the data almost unleverageable and/or ineffective for models. To tackle this scorching challenge, most of the pioneering approaches apply imputation or interpolation in their core, which might lead to embedding data with noise. To especially address this irregular multivariate time series issue, we introduce, in this paper, a new deep neural network model called *ALignment-driven Neural Network*. The innovative idea of our model is to transform the irregular multivariate time series into pseudo-aligned (or pseudo-regular) latent values. The latter are shown as a matrix, where the coefficients are the latent values of each feature at user-defined reference time points that are evenly spaced. They are obtained through a duplication process driven by an exponential decay mechanism. The obtained output is then passed to a Recurrent Neural Network model, which is undoubtedly the must-use model for regular time series data. To show that our model added value, we looked at the Intensive Care Unit mortality prediction task. In this unit, the physiological measurements used to make decisions have a problem with time irregularity. Leveraging the publicly available MIMIC-III, we compare the performance of our model to that of flagship models. In addition, we also performed extensive ablation studies to highlight the importance of specific components in our model. Interestingly enough, whenever data is collected 24 and 48 h after a patient's admission, we outperform our pioneering competitors, i.e., +1.1% and +1.5% for the AUC score, +2.3% and +2.4% for the AUPRC score and +0.6% and +1.7% for the F1-score.

1. Introduction

What justifies the growing use of sensors in various applications (medical, finance, meteorology) is their ability to collect quasi-real-time data associated with various types of content (Choi, Xiao, Stewart, & Sun, 2018; Veillette, Samsi, & Mattioli, 2020; Zebin, Scully, & Ozanyan, 2016). However, due to the wide variety of sensors and data recording methods, it is rare to obtain raw data with all information and input variables sampled simultaneously. Data inconsistency does not systematically reflect a lack of data. In fact, in some applications, the number of times data is sampled may change over time. In this case, the data cannot be considered missing. On the other hand, when data are collected at regular intervals, but some are missing, the series becomes irregular. Furthermore, irregular data can occur for both univariate and multivariate time series. Univariate irregular time series are made up of one characteristic variable measured at a sampling rate with no regular time between observations. For multivariate time series with multiple measurement techniques and instruments, the recording frequency of each variable will often be different. The resulting inconsistent data and missing values make it very difficult to analyze and model the data for tasks such as classification and regression. A simple solution to this problem could be to divide the data collection period into hour-long bins, (El-Rashidy, El-Sappagh, AbuHmed, Abdelrazek, & El-Bakry, 2020; Wanyan, Honarvar, Azad, Ding, & Glicksberg, 2021). However, the disadvantage of this solution is that it eliminates many important

---

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.

E-mail addresses: nzamba.bignoumba@taltech.ee (N. Bignoumba), n.mellouli@iut.univ-paris8.fr (N. Mellouli), sadok.ben@taltech.ee, say@mmmi.sdu.dk (S.B. Yahia).

https://doi.org/10.1016/j.eswa.2023.122148
Received 12 January 2023; Received in revised form 10 October 2023; Accepted 11 October 2023
Available online 16 October 2023
0957-4174/© 2023 Elsevier Ltd. All rights reserved.

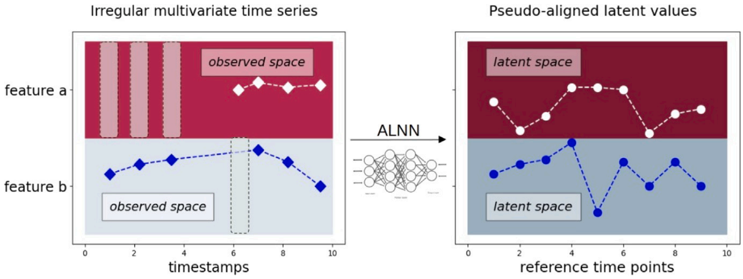


Fig. 1. Graphical representation of the ALNN transformation. The dotted squares on the left plot represent the domains of missing values. On the right plot, at each evenly spaced reference time point, we now have a new latent value for each feature.

data points and exacerbates data loss. In addition, the fact that fine-grained information is lost when aggregation is done in an hour-long bin with many values is another drawback of this method. It can be even worse if an inappropriate aggregation function is used. In other words, this process either removes important fine-grained information, as the granularity of the observed time series may vary from one observation to another depending on the underlying observation context, or introduces noise during the aggregation step.

It is identified that machine and deep learning models are challenged by irregular sampling, which generally assumes fully observed feature representations of an expected fixed size (Shukla & Marlin, 2018). For example, basic RNN models assume that the observation times in a stream are evenly spaced out and that the observation times of different variables for a downstream task are all lined up. However, in real life, time series that are few and far between can rarely meet these assumptions. On one hand, many machine learning and statistical models, e.g., Bayesian Network (MacKay, 1992), Gaussian Processes (Roberts et al., 2013), and Support Vector Regression (SVR) (Vapnik, Golowich, & Smola, 1996), to cite but a few, were applied to address this issue. Still, they failed due to their inability to capture complex temporal dependencies. On the other hand, thanks to their neural architecture dedicated to data extraction and complex pattern detection, deep learning models have shown more promising results (Binkowski, Marti, & Donnat, 2018; Lea, Flynn, Vidal, Reiter, & Hager, 2017; Song, Rajan, Thiagarajan, & Spanias, 2018). Several prediction, classification, or generation tasks involving irregular multivariate time series have seen their accuracy increase thanks to the advent of neural networks. For instance, the implementation of a deep learning model to cope with irregular time series in the laboratory has improved the accuracy of early detection of pancreatic cancer (Park et al., 2022). Xu and Tan (2021) proposed a deep-learning model to efficiently predict asset prices. Tan et al. (2020) proposed a graph-guided neural network for irregularly sampled multivariate time series and showed that their model improved the accuracy of healthcare and human activity classification tasks. Wang, Chen, et al. (2023) developed a deep learning model called BiT-MAC to efficiently impute corrupted data in the ICU[1] and COVID-19 irregular multivariate time series datasets. Tipirneni and Reddy (2022) developed a self-supervised transformer for sparse and irregularly sampled multivariate clinical time series that improves the accuracy of the mortality prediction task. Motivated by these promising results on various tasks, particularly medical tasks, which rely on irregular time series, we decided to propose a deep learning-based solution for one of the most widely studied classification tasks, namely mortality prediction.

What makes mortality one of the most studied classification tasks is that mankind, and researchers in particular, are constantly looking for solutions to reduce the mortality rate. Therefore, being able to predict the risk of mortality will enable healthcare professionals to act upstream to avoid this tragic event (if it can be avoided). Concerned by this issue as humans and as researchers, we decided to propose a deep learning model to predict mortality in intensive care units (ICU) from irregular multivariate time series data. Indeed, due to their temporal properties, physiological measures are part of the electronic health records (EHRs) content widely used in the literature to predict mortality. However, as these are irregular multivariate time series data, processing them is a delicate task. To overcome the aforementioned underlying issues of irregular multivariate time series and make mortality prediction more accurate, we introduce in this paper a deep learning-based model that we called *ALignment-driven Neural Network* (ALNN), which transforms irregular multivariate time series into *pseudo-aligned latent values*. The latter are represented as a matrix, where the coefficients are the latent values of each feature at user-defined reference time points that are evenly spaced. To illustrate this transformation graphically, we can see on the left plot of Fig. 1 that before the ALNN transformation, we had two features whose values were collected at different times. Additionally, the time intervals between the values of these features are irregular. After the ALNN transformation, at each evenly spaced user-defined reference time point, we now have a new latent value for each feature. LANN transformation takes place in latent spaces During this transformation, compared to existing models, e.g., Che, Purushotham, Cho, Sontag, and Liu (2018), Shukla and Marlin (2019a), our model has the advantage of not relying on any in-imputation[2] or in-interpolation[3] that might be noise driver. Instead, this change is made through a duplication process guided by an exponential decay mechanism shown in Fig. 2.

RNNs are ideally designed for regular time series, so they will be more accurate in handling *pseudo-aligned latent values* than irregular multivariate time series. Therefore, the accuracy of the downstream task will improve. It is worth mentioning that ALNN can be used as a stand-alone model for a regression or classification task. In light of this, the significant contributions of the proposed model are as follows:

- We build an ALignment-driven Neural Network (ALNN) to transform irregular multivariate time series into *pseudo-aligned latent values*. In other words, the *pseudo-aligned latent values* are a pseudo-regular version of the initial irregular multivariate time series;
- The ALNN transformation is performed in two stages based on a duplication strategy. In the first stage, after duplicating the value, mask, and time interval matrices as many times as there are reference time points, an exponential decay mechanism is implemented to calculate the time lag penalty score of each value with respect to each reference time point. In the second stage, the time lag penalty score tensor is concatenated with the duplicated

---

[1] Intensive care unit

[2] Imputation performed in the model's core.

[3] Interpolation performed in the model's core.

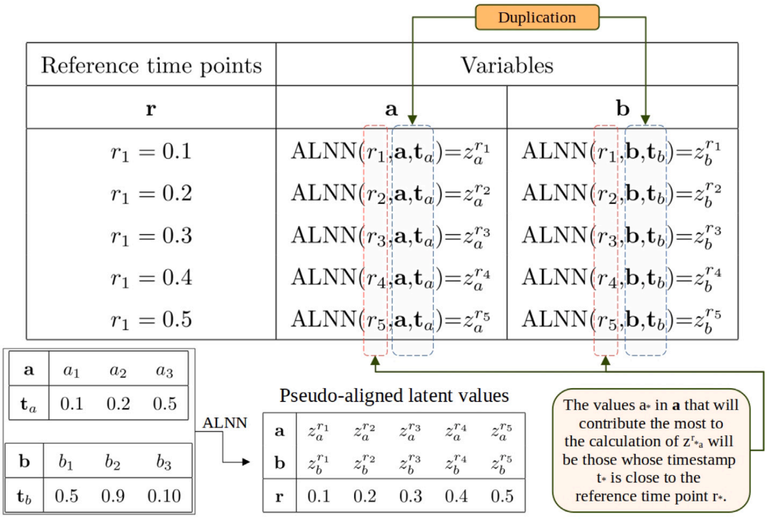

Duplication
Reference time points
Variables
r
a
b
$r_1 = 0.1$
$r_1 = 0.2$
$r_1 = 0.3$
$r_1 = 0.4$
$r_1 = 0.5$
ALNN(r_1,a,t_a)=$z_a^{r_1}$
ALNN(r_2,a,t_a)=$z_a^{r_2}$
ALNN(r_3,a,t_a)=$z_a^{r_3}$
ALNN(r_4,a,t_a)=$z_a^{r_4}$
ALNN(r_5,a,t_a)=$z_a^{r_5}$
ALNN(r_1,b,t_b)=$z_b^{r_1}$
ALNN(r_2,b,t_b)=$z_b^{r_2}$
ALNN(r_3,b,t_b)=$z_b^{r_3}$
ALNN(r_4,b,t_b)=$z_b^{r_4}$
ALNN(r_5,b,t_b)=$z_b^{r_5}$
a a_1 a_2 a_3
t_a 0.1 0.2 0.5
b b_1 b_2 b_3
t_b 0.5 0.9 0.10
ALNN
Pseudo-aligned latent values
a $z_a^{r_1}$ $z_a^{r_2}$ $z_a^{r_3}$ $z_a^{r_4}$ $z_a^{r_5}$
b $z_b^{r_1}$ $z_b^{r_2}$ $z_b^{r_3}$ $z_b^{r_4}$ $z_b^{r_5}$
r 0.1 0.2 0.3 0.4 0.5
The values a* in a that will contribute the most to the calculation of z^r*a will be those whose timestamp t* is close to the reference time point r*.

Fig. 2. Duplication process applied in ALNN. **a** and **b** are two univariate time series with different timestamps $\mathbf{t}_a$ and $\mathbf{t}_b$, respectively. The time interval between the values of $\mathbf{t}_a$ and $\mathbf{t}_b$ is irregular. **r** is the user-defined vector of reference time points. In contrast to $\mathbf{t}_a$ and $\mathbf{t}_b$, the time interval $\Delta r = 0.1$ in **r** is regular. **a** and **b** and their respective timestamps $\mathbf{t}_a$ and $\mathbf{t}_b$, are duplicated as many times as there are reference time points. The pseudo-aligned latent value matrix, which is the global ALNN output, contains the new latent values $z_*^{r_*}$ of **a** and **b** at different reference time points. Note that to keep the illustration simple, we only implied **a**, **b**, $\mathbf{t}_*$, and **r** as ALNN parameters. However, as we will see in Section 3, other matrices are involved in the calculation. The principle for calculating $z_*^{r_*}$ will remain the same (read the square's contents at the bottom right).

version of the value, mask, and time interval matrices to produce the pseudo-aligned latent values via an alignment process;

- The advantage of the proposed model is that no potentially noisy in-imputation/interpolation is required to deal with the temporal irregularity. Furthermore, the fact that the value of each feature is latent makes the downstream task more accurate. Indeed, although latent values are less explicable than observed values, they are nevertheless more informative in the sense that they encode relevant hidden patterns;
- We combine ALNN with GRU (Dey & Salem, 2017) (ALNN-GRU) to predict mortality in the intensive care unit. The training is performed in an end-to-end fashion to jointly optimize the parameters of both models;
- We validated the proposed model using the MIMIC-III database (Johnson et al., 2016). After a thorough empirical evaluation, we found that our proposal sharply outperforms state-of-the-art models designed to handle multivariate time series data.

The remainder of this paper is organized as follows: We present the related works in Section 2. A formal description of our model is presented in Section 3. In Section 4, we discuss the performance of the ALNN-GRU on the pilot case, which is the mortality prediction task, and compare it *versus* the state-of-the-art models. We also perform ablation studies to show the importance of specific model components. Section 5 discusses some model limitations and how they can be addressed. Section 6 reminds the takeaways and contributions of this paper and sketches pathways for future work.

2. Related work

This paper pays heed to the research on the scorching issue of irregular data from univariate and multivariate systems, where records with irregular sampling intervals and a timestamp do not have values for every variable in the feature space. In terms of the amount of missing data, there is a snug connection between data sparsity and data irregularity. In real-world time series datasets, the amount of missing data can vary greatly from one domain to the next. For example, in the case of low-frequency data, environmental samples may contain fewer than 10% missing observations. In contrast, for high-frequency data such as intensive care unit (ICU) data, samples can commonly contain 80% missing data in a multivariate feature space, while the sparsity of financial transactions can be interpreted as extremely high since transactions in multiple stocks very rarely occur at the same time and transactions are made regularly. Many common statistical models, like ARIMA (Chen, Wang, & Huang, 1995), or Gaussian processes (Roberts et al., 2013), and many classic machine learning models, like KNN (Martínez, Frías, Pérez, & Rivera, 2019) or SVM (Vapnik et al., 1996), have also been used a lot to solve time series problems. However, these methods cannot capture the complex temporal relationships between observations in univariate and multivariate time series. Recurrent Neural Networks (RNNs) have been shown to be more efficient in dealing with regular multivariate time series. In particular, gated RNNs and their scalable architectures are among the most widely used methods for time series modeling to date. Since modeling time series data with gated RNNs has been successful, our paper contributes to further improving these models to handle irregular time series data.

The remainder of this section is dedicated to the scrutiny of RNN-based and non-RNN-based approaches.

2.1. RNN-based model

Dealing with multivariate time series data, also known as sporadically observed time series, comes down to defining an approach to impute or interpolate missing values caused by temporal irregularity and other aspects, such as not collecting data during a period of time t. One of the most common and straightforward acceptable approaches used in El-Rashidy et al. (2020), Wanyan et al. (2021) is to discretize the time period over which the data are collected, aggregate values belonging to the same hour-long bin, and perform an imputation (with 0, median, or mean) on timestamps without observed values. Still, this imputation does not allow the model to distinguish between imputed and non-imputed values. Instead, it grants the same level of trust to both. Then, the authors in Lipton, Kale, and Wetzel (2016) proposed a continuous-time observation discretization into an hour-long bin. They introduced a mask matrix, allowing their model to capture the missing informative missing patterns. To capture the level of uncertainty of imputed values and propagate it over time, the authors in Jun, Mulyadi, Choi, and Suk (2021) proposed an uncertainty-gated stochastic sequential model combined with a recurrent variational network that estimates the distribution of missing values. In Yoon, Zame, and van der Schaar (2019), the authors proposed a multi-directional recurrent neural network to interpolate and impute between data streams while capturing the uncertainty of the imputed values. Although these models work well in practice, they require ad-hoc time bin management and could provide missing data when empty bins are detected or lose data when fine-grained information is removed. Our proposal overcomes this issue by directly dealing with the raw, irregular multivariate time series. Ad-hoc time bin management is no longer necessary.

To avoid this ad hoc time bin management, several models have used a data-driven imputation approach. A data-driven approach steers the imputation partially (or totally) with the downstream objective loss function. For instance, a gated recurrent network incorporating a decay mechanism (GRU-D) that imputes missing values to its core was introduced in Che et al. (2018). The missing value is imputed by capturing the long-term temporal dependencies in the time series via a decay mechanism conditioned on a time interval matrix. A mask matrix is also incorporated to identify the values that need to be imputed and capture missing patterns. Unlike the GRU-D, whose imputation process relies on a medical assumption, the authors in Cao et al. (2018a) proposed a more generic imputation model. The latter is a modified bi-directional RNN in which recurrent data-driven imputation is performed while accounting for the correlation between variables. In addition to a recurrent imputation that exploits the correlation between variables, the authors in Suo et al. (2020) combine the latter with another recurrent imputation that captures specific patterns of individual variables. Subsequently, they implement multidirectional self-attention to learn long-term dependencies across time and variables. In the same trend, the authors in Shukla and Marlin (2019b) proposed a semi-parametric interpolation network that also performed imputation in its core. Therefore, they propose to interpolate the missing values based on several radial basis functions (RBFs) concerning a set of reference points defined on the timescale. The aim of this interpolation is to generate regular time series from irregular data. Similarly, Tan et al. (2021) proposed an explainable uncertainty-aware convolutional recurrent neural network for irregular medical time series, in which they use a Gaussian process to generate regular data from irregular ones and estimate the uncertainty of the generated values. In addition, a hierarchical uncertainty-aware decomposition is implemented to adaptively decompose the time series into different subseries and assign them appropriate weights based on their reliability. Since the approach of imputing values with respect to a set of evenly spaced reference time points may suppress the underlying information conveyed by the temporal irregularity and affect the accuracy of the downstream task, Tan et al. (2020) proposed a GRU-based model called DATA-GRU that imputes missing values only with respect to observed timestamps. This imputation is performed by a time-aware mechanism that also produces reliability vectors of the imputed values. A dual-attention mechanism is then implemented to deal with missing values by jointly considering data quality and medical knowledge. The disadvantage of data-driven imputation/interpolation models that do not rely on evenly spaced reference points but instead intersect all timestamps of all features to obtain a single vector of timestamps is that this intersection increases the length of the input and therefore the number of parameters. To overcome this problem, we favor the use of evenly spaced reference time points. However, to reduce the noise that may be introduced by the latter, we parameterize the time interval between them in order to be able to select the least noisy reference time points using a technique such as grid search.

Data-driven imputation/interpolation models calculate new values, which are combined with the initial values to make the decision. However, these new values may be noise factors and then alter the hidden structure of the data. Instead, with the duplication process, only the initial values are used in the model calculation. A similar approach of relying only on initial value has been proposed by Neil, Pfeiffer, and Liu (2016). They proposed a Phased-LSTM, which is a variant of LSTM with an additional time gate. The units of this time gate process a set of signals that control the updating of hidden and memory cells. The hidden and memory cells do not need to be updated at each observed time, which is substantial in terms of efficiency.

2.2. *ODE-based model*

Unlike classical RNNs consisting of discrete hidden layers, which makes them intrinsically suitable for regular time series, alternative deep learning models based on ordinary differential equations (ODE) have also been proposed. Their particularity is to process irregular time series in a continuous latent space. In Chen, Rubanova, Bettencourt, and Duvenaud (2018), the precursors of this approach proposed substituting the sequence of hidden layers with a continuous latent function relying on an ODE black box that captures the continuous-time dynamic flow. Inspired by Chen et al. (2018), the authors Rubanova, Chen, and Duvenaud (2019) proposed to combine ODE with RNNs. Their approach is to compute the hidden states of an RNN via ODE conditioned on a continuous latent function, the previous hidden state, and the current and previous timestamp values. Rather than using RNNs that suffer from vanishing or exploding gradient problems during training, the authors in Lechner and Hasani (2020) combined ODE with an LSTM designed to address the vanishing problem. The LSTM then makes it possible to separate the memory unit from its latent time-continuous state, which might introduce a vanishing or exploding gradient problem. A similar work, based on ODE, was also carried out in Kidger, Morrill, Foster, and Lyons (2020). Although ODE-based models have the particularity of processing time series in a continuous space and therefore overcome the problem of irregular time intervals, these models have the disadvantage of requiring more computation time. Moreover, calculating latent values at irrelevant time points may introduce noise into the learning process, affecting the downstream task's accuracy. Since ODE processes time series in a continuous time–space, there is no way to avoid including irrelevant time points in the learning process. However, with our model, thanks to the parameterization of the time interval between reference points, we can find a snug approximation of the behavior of a continuous function (if necessary), which reduces the rate of irrelevant time points in the learning process.

2.3. *Transformer-based model*

More recently, due to their effectiveness over classical models such as RNN in NLP tasks, several researchers have implemented transformer-based models to process irregular time series data (Shan, Li, & Oliva, 2021; Tipirneni & Reddy, 2022; Wang et al., 2021). Indeed, thanks to the position encoding component, they were able to capture irregular temporal patterns. For example, to avoid any imputation or interpolation of missing values caused by the irregularity within the time series, Lee, Jun, and Suk (2021) proposed a multi-view integration approach relying on a time interval matrix, mask, and observed value matrices. The latter integrates the missingness information by processing the time interval and mask matrices with a multi-head attention component. Du, Côté, and Liu (2023) developed a self-attention-based model called SAITS to impute missing values in multivariate time series. The self-attention applied in this model is enhanced by two diagonally-masked self-attention (DMSA) for better capture of feature correlation and temporal dependencies. SAITS is trained using a joint optimization approach. Although SAITS produces functional results, there is no evidence that it will work correctly

Table 1
Competing models with the techniques or approaches they use to deal with irregular time series. Our model is also listed. $\bar{\times}$ stands for pseudo. GRU-mask is a GRU whose inputs are a concatenation of value matrices and mask matrices.

RNN-based	ODE-based	Competing models	Discretization	In-imputation/ interpolation	Alignment mechanism	Irregular time modeling	Continuous time modeling	Missing information modeling	Duplication
×		BRITS (Cao et al., 2018b)		×		×		×	
×		Interp-net (Shukla & Marlin, 2019a)		×	×	×			
×		GRU-D (Che et al., 2018)		×		×		×	
×		GRU-mask (see Section 4.6.1)	×					×	
×	×	ODE-LSTM (Lechner & Hasani, 2020)					×		
×		mTAND (Shukla & Marlin, 2021a)		×	×			×	
	×	Neural-CDE (Chen et al., 2018)					×		
×		Phased-LSTM (Neil et al., 2016)				×			
×		ALNN-GRU (Neil et al., 2016)			×	×	$\bar{\times}$	×	×

with irregular multivariate time series. Narayan Shukla and Marlin (2021) built a multi-time attention network composed of an inference network (encoder) and a generative model (decoder). Technically, in the encoder, a set of latent variables is picked from a learned distribution based on the output of multi-time attention blocks and a set of reference time points for a classification task. On the other hand, the decoder figures out what the missing values should be when asked for. However, in this approach, which is probably the closest to ours, the imputation was made without considering the heteroscedastic aspect. To address this, Shukla and Marlin (2021b) proposed a heteroscedastic temporal variational autoencoder model.

Although there is great potential for applying transformers to time series data, current research in this area is still limited, and the results are mitigated. Indeed, the memory intensity of standard transformers is one of the limitations of very long sequences and their quadratic time complexity compared to the linear complexity of RNNs. Besides, processors still have difficulty extracting features due to their lack of true recurrent gradients and inability to encode positional information faithfully. These shortcomings with transformers compared to the ongoing evolution of gated recurrent networks demonstrate a strong interest in using gated RNN models for time series data.

Attention-based mechanisms have been approved to make logs more sensitive to their local contexts and more aware of them, as well as to make standard transformers less complicated.

2.4. Graph-based model

Graph neural networks have attracted a great deal of interest in time series modeling because of their flexibility and ability to capture spatio-temporal dependencies between data points and features. For instance, Zhang, Zeman, Tsiligkaridis, and Zitnik (2022) proposed a graph neural network called RAINDROP, in which the vertices represent the sensors (features) and the edges represent the relationships between them. The principle of this method is to map each sample, which is modeled as a graph, into an embedding vector of fixed dimension. Message passing, which models the relationship between sensors, is used to estimate the embedding value of the sensors when they are not observed at a given timestamp. Chen, Ding, and Zhai (2022) proposed decomposing each univariate time series into different intrinsic mode functions (IMFs) and residuals using the Empirical Mode Decomposition (EMD) and modeling each IMF and residual as nodes of a graph neural network. A multi-head attention mechanism and a temporal convolutional network (TCN) are then used to learn the correlation between nodes and encode temporal relationships, respectively. Although the combination of these different models has proved effective for some regression tasks involving regular multivariate time series, there is no evidence of their effectiveness for tasks involving irregular multivariate time series. As the dynamics of multivariate time series can change over time, Li, Yu, Zhang, and Xu (2023) proposed a dynamic graph neural network that models this change. Whether in the learning or testing phase, the dynamic properties of the graph allow it to rebuild itself in the event of a change detected in the input data. Oskarsson, Sidén, and Lindsten (2023) built a temporal graph neural network for irregular data, in which they introduced a time-continuous latent state in each node (observation). Similar works implementing graph neural network-based models for irregular time series modeling were conducted in Cini, Marisca, and Alippi (2021), Wang, Liu, et al. (2023).

The limitation of graphical neural networks is that they may require expert assistance to construct the links in the graph. As for those that implement data-driven link construction, they are often designed for regular multivariate time series.

2.5. Takeaway messages of the scrutiny of the related work

Table 1 summarizes common approaches and techniques used in models that deal with irregular time series. We have limited this summary to our model and those of our competitors. These approaches and techniques are as follows:

- **Discretization**: discretization of the time period over which data are collected;
- **In-imputation/interpolation**: the imputation or interpolation of the missing values is carried out in the model's core and is driven by the objective loss function(s) related to the downstream task;
- **Alignment mechanism**: A set of ordered reference time points is used to either estimate or interpolate missing values at these points or to calculate the latent values for each. This allows obtaining values (observed or latent) with the same time trend in and across streams;
- **Irregular time modeling**: The model deals with raw, irregular time series. No discretization or alignment mechanism is implemented. They often rely on decay functions to model the irregularity of time;
- **Continuous time modeling**: the model processes the irregular time series with a continuous-time function(s) rather than using a sequence of discrete hidden layers;
- **Missing information modeling**: Strategies implemented in models' core to differentiate between observed and missing values. The most common technique is the use of a binary mask to indicate whether a value is observed or missing (or imputed) (*c.f.* Eq. (2)).
- **Duplication**: duplication of values to fill in the reference time points. This duplication is guided by an exponential decay mechanism that weights each value according to its temporal distance from each reference time point.

In view of the aforementioned drawbacks faced by the existing models, we introduce our new model, called ALNN-GRU. As highlighted by the last row of Table 1 the sighting items of ALNN-GRU are as follows:

- We process the raw irregular multivariate time series (**irregular time modeling**) to preserve the fine-grained information and underlying structure of the data. In doing so, we avoid the pitfall of performing any **discretization** or **in-imputation \interpolation** that might remove fine-grained information or introduce more noise;
- As we impute the data for the reasons listed in Section 4.3, we introduce a binary mask, which will make the model less focused on imputed values (**missing information modeling**);
- We implement a **duplication** process that relies solely on the initial values to avoid introducing noisy values into the model calculation.
- We implement an **alignment mechanism** to provide the RNN, which works best with regular time series, with a "regularized" version of irregular multivariate time series.
- By relying on configurable reference time points, our model can approximate the behavior of a continuous function (**continuous time modeling**).

3. Method

Basic RNN models assume a regular time interval between observations in the case of univariate time series and aligned data points in the case of multivariate time series. Aligned data refers to the fact that the values across features are all collected following the same time trend. Therefore, feeding an RNN model with irregular univariate or multivariate time series observations would undeniably worsen performance. To address this issue, we build on top of the latter a deep-based preprocessing model called ALNN that aims to transform irregular multivariate time series data into pseudo-regular multivariate latent time series, which we have named "*pseudo-aligned latent values*".

Before thoroughly elaborating on ALNN-GRU, we present the preliminary mathematical notations, definitions, and hypotheses necessary for a smooth understanding of our proposal in the following subsections.

3.1. Preliminaries

We first detail the mathematical notations, then the key definitions of our approach, and finally, present the different hypotheses on which our approach is grounded.

3.1.1. Mathematical notations

We let $\mathcal{D} = \{(X_n, M_n, T_n, \Delta_n, y_n)_{n=1,2,\ldots,N}\}$ represents a dataset, where N is the number of samples. $X_n = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K]^\top$ is the multivariate time series, where $\mathbf{x}_k \in \mathbb{R}^{j}_{(j=1,2,\ldots,J)}$ is the kth univariate time series. As the number of values j in the streams may differ, we fixed a length J for all streams. J is set following a process described in Subsection 4.4. Thus, $X_n(x_j^k) \in \mathbb{R}^{J\times K}$ and x_j^k is either observed or imputed. $T_n(t_j^k) \in \mathbb{R}^{J\times K}$ is the timestamp matrix, $M_n \in \{0,1\}^{J\times K}$ with coefficients m_j^k, is a binary matrix that indicates whether a value in X_n has been imputed or not. $\Delta_n(\delta_j^k) \in \mathbb{R}^{J\times K}$ is the time interval (or variation) matrix where columns represent the time intervals between the values of each stream (univariate time series), and y_n is a single target whose value is discrete in the case of classification and real-valued in the case of regression. In the case of mortality prediction, the target y_n value is a binary value, i.e., 1 for dead, 0 for alive. The remaining notations are described in Table 2.

Example 1. We have a multivariate time series represented by X shown below, where the subscript n has been dropped to simplify notation. Three values are collected, and three are imputed (X imputed values are in bold).

$$X = \begin{bmatrix} 1.5 & \mathbf{80.0} & \mathbf{67.5} \\ 1.9 & \mathbf{80.0} & 70.0 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 4 \end{bmatrix} \tag{1}$$

Table 2
ALNN-GRU notations.

Notation	Definition
$\mathbf{X}, \mathbf{M}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{I}$	Tensors of observed values, masks, timestamps, time intervals, and penalties.
X, M, T, Δ, I, Z	Matrices of observed values, masks, timestamps, time intervals, penalties, and pseudo-aligned latent values.
$\mathbf{r}, \mathbf{z}$	Reference time points and latent global state of the patient (also called context vector).
J, K, P	The number of values per feature, Number of features, and the number of reference time points.
Q	Is a condensed representation of $P \times J \times K$.
N'	It represents the batch.
$\langle . \rangle, \sigma(.), \vert . \vert$ $exp(.), \phi(.), \odot$	Concatenation symbol, activation function, absolute distance, exponential function, sum function, and Hadamard product.
x_j^k, m_j^k, t_j^k and δ_j^k	They are respectively the coefficients of $X, M, T\ \Delta$.
$(i_j^k)_p$	$(i_j^k)_p \in I_p \in \mathbf{I}$. It represents the penalty score of x_j^k at the reference time r_p.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad m_j^k = \begin{cases} 1 & \mathrm{i}f\ x_j^k\ is\ observed \\ 0 & \mathrm{o}therwise \end{cases} \tag{2}$$

$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 3 \end{bmatrix} \qquad \delta_j^k = \begin{cases} t_j^k - t_{j-1}^k + \delta_{j-1}^k & \mathrm{j} > 1,\ \mathrm{m}_{j-1}^k = 0 \\ t_j^k - t_{j-1}^k & \mathrm{j} > 1,\ m_{j-1}^k = 1 \\ 0 & \mathrm{j} = 1 \end{cases} \tag{3}$$

The imputed values are due to outliers, streams of different lengths, or streams with no values. The first value of the first feature ($k = 1$), $x_1^1 = 1.5$, is the full value, and we assign 1 to m_1^1. It is observed at the timestamp $t_1^1 = 1$ (in hours). The time interval between the observed values $x_1^1 = 1.5$ and $x_2^1 = 1.9$ is $\delta_2^1 = 1$. It should be remembered that the time interval (δ) between successive timestamps is not constant for an irregularly sampled dataset.

3.1.2. Definitions

In this subsection, we define two key concepts.

Reference time points. They are ordered and evenly spaced values $r_p \in \mathbf{r} = [r_1, r_2, \ldots, r_P]_{p=1,\ldots,P}$ that represent a temporal discretization of the period during which we collect the data. For example, if we decide to collect data for the first 5 h, $\mathbf{r} = [0, 1, 2, 3, 4, 5]$ and the time interval $\Delta r = 1$. Notably, $\mathbf{r}$ and Δr are user-defined hyperparameters.

Pseudo-aligned latent values. They are transformations of irregular multivariate series that ALNN takes as input. They are represented in matrix form. Each coefficient in each row corresponds to the latent value of a feature k at a reference time point r_p.

3.1.3. Hypotheses

This subsection presents the underlying assumptions of our ALNN-GRU model.

Hypothesis 1. We assume that the perfect time interval Δr between the reference time points r_p can be obtained by a grid search performed by the user rather than being a data-driven interval;

Hypothesis 2. Since some streams may not have the number J of values set per stream, we assume that these missing values and their corresponding timestamps can be imputed by simply repeating the last value and timestamp of the stream until we obtain J values;

Hypothesis 3. We suppose that the further the timestamp value t_j^k of x_j^k is temporally from a reference time point r_p, the less influence it has at this reference time point.

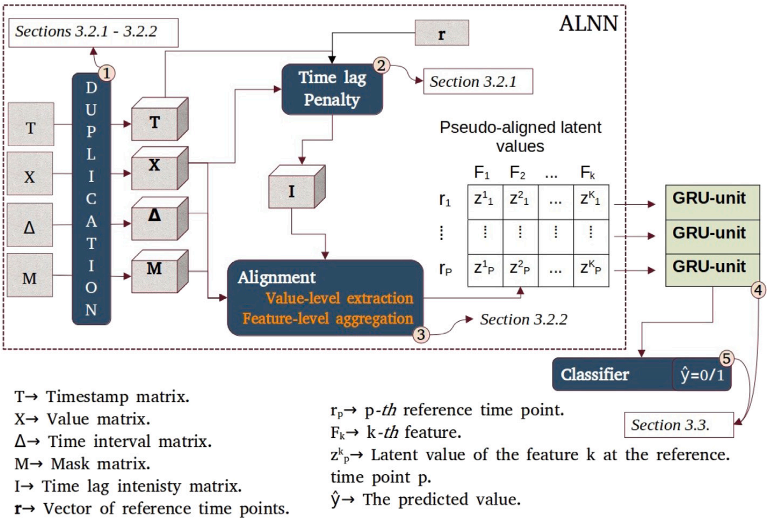


Fig. 3. Architecture of the ALNN-GRU.

After introducing all the notations, definitions, and hypotheses, we formally describe ALNN in the next subsection.

The final model, whose overall architecture is glanced at in Fig. 3, is called ALNN-GRU since it combines ALNN and RNN models, specifically a GRU. A glance at this figure shows that the ALNN performs an initial preprocessing stage on irregular multivariate time series inputs before passing them through the RNN. Unlike the initial multivariate time series, which suffers from temporal irregularity, the *pseudo-aligned latent values* can be more efficiently handled by an RNN model to obtain a context vector. The latter is then passed sequentially to the last layer, which plays the classifier (*resp*. generator) role in a classification (*resp*. regression) task.

3.2. ALNN-GRU-step 1: The alignment-driven neural network (ALNN)

The main goal of the neural network ALNN is to transform the irregular multivariate time series into *pseudo-aligned latent values*. The ALNN transformation is performed in two steps, namely the computation of the time lag penalty scores and the alignment process. Both steps are thoroughly described in the following.

3.2.1. Time lag penalty

The Time lag Penalty step aims to compute the time lag penalty score of all values, i.e., their influence at each reference time point. Roughly speaking, the time lag penalty score indicates the amount of information to be considered from x_j^k given a reference time point r_p. This penalty score is computed following Hypothesis 3 of Section 3.1.3. The penalty score matrix I is calculated as follows:

$$I_p = X \odot exp\{-max(0, -\gamma_p|r_p - T|)\}; I_p \in \mathbb{R}^{J\times K} \tag{4}$$

where the values $(i_j^k)_p \in I_p$ are the penalties score of each value x_j^k at the reference time point r_p. γ_p is a value of the transition vector $\Gamma = [\gamma_1, \ldots, \gamma_P]$. Each value, γ_p, is associated with a reference time point r_p. Γ is a learnable parameter. $|r_p - T|$ is the absolute distance between timestamps t_j^k of x_j^k and a reference time point r_p. As a penalty score matrix is needed for all reference time points, it is then computed P times. The global penalty score tensor is as follows:

$$\mathbf{I} = \left[\begin{array}{ccc} I_1, & \cdots & , I_P \end{array}\right]^{\mathsf{T}}; \mathbf{I} \in \mathbb{R}^{N'\times Q} \tag{5}$$

N' is the batch size value, which is set to 1 for the simplicity of matrix representations in Appendix A, and $Q = P \times J \times K$. The rationale behind choosing the absolute distance rather than the Euclidean distance is that the variation between the timestamps t_j^k and the reference time points r_p will generally be high. The absolute distance is more appropriate for high variability, while the Euclidean distance is for low variability.

In the following subsection, we formally describe how the penalty tensor score $\mathbf{I}$ is integrated into the alignment component to transform the irregular multivariate time series into *pseudo-aligned latent values*.

3.2.2. Alignment

In the following, we describe how the multivariate time series X is transformed into a *pseudo-aligned latent values* through a duplication process driven by the penalty score tensor $\mathbf{I}$. The duplication process (*c.f.* Eq. (6)) consists of duplicating X, M and Δ P times. This allows each set $[X, M, \Delta]$ to be concatenated with a penalty matrix score I_p for value-level extraction and feature-level aggregation, explained below and illustrated in Fig. 4.

$$\mathbf{X} = [\underbrace{X, \ldots, X}_{1,\ldots,P}]^{\mathsf{T}},\ \mathbf{M} = [\underbrace{M, \ldots, M}_{1,\ldots,P}]^{\mathsf{T}},\ \boldsymbol{\Delta} = [\underbrace{\Delta, \ldots, \Delta}_{1,\ldots,P}]^{\mathsf{T}} \tag{6}$$

It is worth mentioning that we favor the duplication approach over the iterative approach for parallel computing. This strategy is time-effective.

Value-level extraction. It consists of calculating an embedded value $(v_j^k)_p \in \mathbf{V}$, which is a non-linear combination of x_j^k, m_j^k, δ_j^k and $(i_j^k)_p$. In other words, $(v_j^k)_p$ is a faithful latent representation of x_j^k carrying the missingness, temporal transition, and time lag penalty information. The embedded value $(v_j^k)_p$ is computed as follows:

$$(v_j^k)_p = \sigma(x_j^k(\hat{w}_1^k)_j^p + m_j^k(\hat{w}_2^k)_j^p + \delta_j^k(\hat{w}_3^k)_j^p + (i_j^k)_p(\hat{w}_4^k)_j^p + (\hat{b}^k)_j^p) \tag{7}$$

$\sigma(.)$ is an activation function. $(\hat{w}_{[1,4]}^k)_j^p \in \hat{W} \in \mathbb{R}^{1\times Q\times 4}$ and $(\hat{b}^k)_j^p \in \hat{B} \in \mathbb{R}^{1\times Q\times 1}$ are learnable parameters. The global matrix of all embedded values $(v_j^k)_p$ is then obtained as follows:

$$\mathbf{V} = \sigma(\phi(< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} > \odot \hat{W} + \hat{B}));\ (v_j^k)_p \in \mathbf{V} \in \mathbb{R}^{N'\times Q\times 1} \tag{8}$$

where $\phi(.)$ is the sum function of the coefficients of $(< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} > \odot \hat{W} + \hat{B}) \in \mathbb{R}^{N'\times Q\times 4}$ along the last axis (*c.f.* Eq. (7)). $\odot$ is the Hadamard product. We use the Hadamard product at this level to keep the calculation of each embedding value $(v_j^k)_p$ separate. Formally, $(v_j^k)_p$

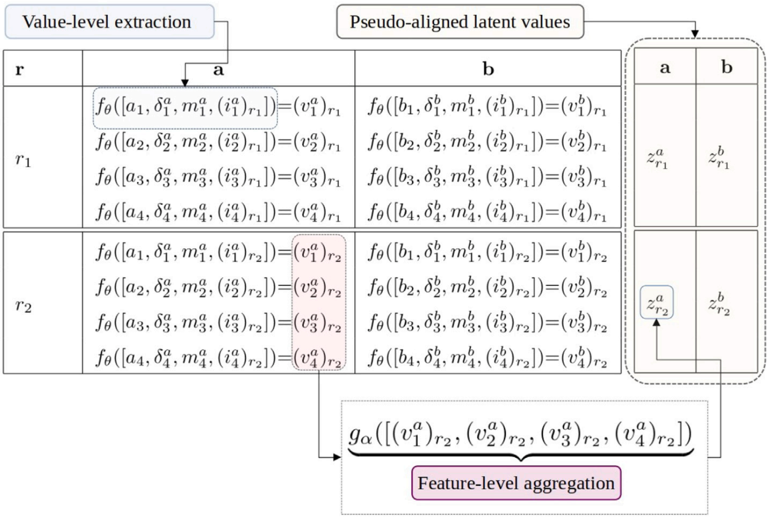

Value-level extraction
Pseudo-aligned latent values
r
a
b
r_1
$f_\theta([a_1,\delta_1^a,m_1^a,(i_1^a)_{r_1}])=(v_1^a)_{r_1}$
$f_\theta([a_2,\delta_2^a,m_2^a,(i_2^a)_{r_1}])=(v_2^a)_{r_1}$
$f_\theta([a_3,\delta_3^a,m_3^a,(i_3^a)_{r_1}])=(v_3^a)_{r_1}$
$f_\theta([a_4,\delta_4^a,m_4^a,(i_4^a)_{r_1}])=(v_4^a)_{r_1}$
$f_\theta([b_1,\delta_1^b,m_1^b,(i_1^b)_{r_1}])=(v_1^b)_{r_1}$
$f_\theta([b_2,\delta_2^b,m_2^b,(i_2^b)_{r_1}])=(v_2^b)_{r_1}$
$f_\theta([b_3,\delta_3^b,m_3^b,(i_3^b)_{r_1}])=(v_3^b)_{r_1}$
$f_\theta([b_4,\delta_4^b,m_4^b,(i_4^b)_{r_1}])=(v_4^b)_{r_1}$
r_2
$f_\theta([a_1,\delta_1^a,m_1^a,(i_1^a)_{r_2}])=(v_1^a)_{r_2}$
$f_\theta([a_2,\delta_2^a,m_2^a,(i_2^a)_{r_2}])=(v_2^a)_{r_2}$
$f_\theta([a_3,\delta_3^a,m_3^a,(i_3^a)_{r_2}])=(v_3^a)_{r_2}$
$f_\theta([a_4,\delta_4^a,m_4^a,(i_4^a)_{r_2}])=(v_4^a)_{r_2}$
$f_\theta([b_1,\delta_1^b,m_1^b,(i_1^b)_{r_2}])=(v_1^b)_{r_2}$
$f_\theta([b_2,\delta_2^b,m_2^b,(i_2^b)_{r_2}])=(v_2^b)_{r_2}$
$f_\theta([b_3,\delta_3^b,m_3^b,(i_3^b)_{r_2}])=(v_3^b)_{r_2}$
$f_\theta([b_4,\delta_4^b,m_4^b,(i_4^b)_{r_2}])=(v_4^b)_{r_2}$
a
b
$z_{r_1}^a$
$z_{r_1}^b$
$z_{r_2}^a$
$z_{r_2}^b$
$g_\alpha([(v_1^a)_{r_2},(v_2^a)_{r_2},(v_3^a)_{r_2},(v_4^a)_{r_2}])$
Feature-level aggregation

Fig. 4. Illustration of the alignment process. **a** and **b** are two features and $\mathbf{r} = [r_1, r_2]$ the reference time points vector. f is the nonlinear function that performs the values-level extraction and $\theta = \{\hat{W}, \hat{B}\}$ its parameters (*c.f* Eq. (7),(8)). g is the nonlinear function that performs the feature-level extraction and $\alpha = \{\bar{W}, \bar{B}\}$ its parameters (*c.f* Eq. (9),(10)).

is uniquely obtained from x_j^k, its corresponding mask, time interval, and time lag penalty. All $x_{j'}^{k'} \neq x_j^k$ and their respective mask, time interval, and time lag penalty are ignored ($k' \in [0, K]; j' \in [0, J]$).

Once the values-level extraction is achieved, we perform the feature-level aggregation.

Feature-level aggregation. It consists of merging all embedded values $(v_{:J}^k)_p$ by weighting each of them. This merging process, followed by a nonlinear transformation, aims to obtain a single latent value z_p^k (Eq. (10)) for the feature k at the reference time point r_p. Therefore, the *pseudo-aligned latent values* Z, which is the global matrix of all z_p^k, is obtained as follows:

$$Z = \sigma(\phi(\bar{\mathbf{V}} \odot \bar{W}_z + \bar{B}_z)); Z \in \mathbb{R}^{P\times K} \tag{9}$$

where $\sigma(.)$ is an activation function; $\phi(.)$ the sum function of the coefficients of $(\bar{\mathbf{V}} \odot \bar{W}_z + \bar{B}_z) \in \mathbb{R}^{N'\times Q}$ along the third axis ($N' = 1$ and $Q = P \times J \times K$). $\bar{V}(\bar{v}_j^k)_p \in \mathbb{R}^{N'\times Q}$ is the reshaped version of $V \in \mathbb{R}^{N'\times Q\times 1}$. $\bar{W}_z((\bar{w}_j^k)^p) \in \mathbb{R}^{1\times Q}$ and $\bar{B}_z((\bar{b}^k)^p) \in \mathbb{R}^{1\times P\times K}$ are model parameters. The Hadamard product allows weighing the latent values of a feature k without considering the other features' values. Unlike some works, such as Wang et al. (2021), we do not consider the correlation of variables to address the irregularity issue. The coefficients z_p^k of the *pseudo-aligned latent values* Z are calculated as follows:

$$z_p^k = \sigma\left(\sum_{i=1}^{J} (\bar{v}_i^k)_p(\bar{w}_i^k)^p + (\bar{b}^k)^p\right); z_p^k \in Z \tag{10}$$

Remark 1.

1. The term *aligned* refers to the fact that in Z, each row $z_p^{:K}$ represents the latent values of each feature k at the same reference time point r_p. Whereas it is unlikely for X, values in the same row have different timestamps.
2. The term *pseudo* was introduced since the calculation of z_p^k does not only rely on all embedded representations $(v_{:J}^k)_p$ of all $x_{:J}^k$ with the timestamp $t_j^k = r_p$. Rather, it depends on all embedded representations $(v_{:J}^k)_p$ of $x_{:J}^k$ having timestamps t_j^k such as $r_1 <= t_j^k <= r_P$. In other words, z_p^k, the new feature value k at the reference time point r_p is calculated by considering all values x_j^k observed before and after the reference time point r_p (*c.f* Fig. 4 for illustration).

Remark 2. We want to emphasize that although our duplication process may introduce some noise, it remains negligible. Indeed, even though the reference time interval delta is small $\Delta r = \epsilon$, the difference between the penalty scores of x_j^k at the reference time points r_p and $(r_p + \epsilon)$ will "most often" be non-zero, thanks to the exponential function. We said "most often" because this difference can be zero due to our computers' finite number of bits. In this case, noise is introduced.

So far, no sequential pattern between rows of Z has been captured. The following subsection presents how *pseudo-aligned latent values* are passed through a GRU for sequential pattern detection and sequentially into a Dense layer for the final binary classification task. It is worth mentioning that the final task can also be any regression, multi-regression, or multi-classification task.

3.3. ALNN-GRU-step 2: The prediction model

We used a GRU, an RNN model that solves the vanishing problem, to take advantage of the sequential pattern of *pseudo-aligned latent values*. The output $\mathbf{z}$ of the GRU named context vector, seen as a global latent representation of a patient over the r_P hours, is fed sequentially into a dense layer for the mortality prediction task. Formally:

$$\mathbf{z} = GRU_\theta(Z) \tag{11}$$

$$\hat{y} = Dense_\alpha(Sigmoid(\mathbf{z})) \tag{12}$$

$$Sigmoid(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}} \tag{13}$$

GRU and Dense parameters are denoted by θ and α, respectively. $Sigmoid(.)$ represents the activation function. We used it because our downstream task is a binary classification task. Other activation functions, such as $Linear(.)$ or $Softmax(.)$, could be used for regression or multiple classification tasks. $\hat{y} \in [0, 1]$ represents the likelihood that the patient will die.

As far as we are dealing with unbalanced data, i.e., the major class "0" is by far more numerous than the minor class "1", we used the focal loss (Lin, Goyal, Girshick, He, & Dollár, 2017) to cope with such an issue. After extensive experiments, the focal loss was selected, and

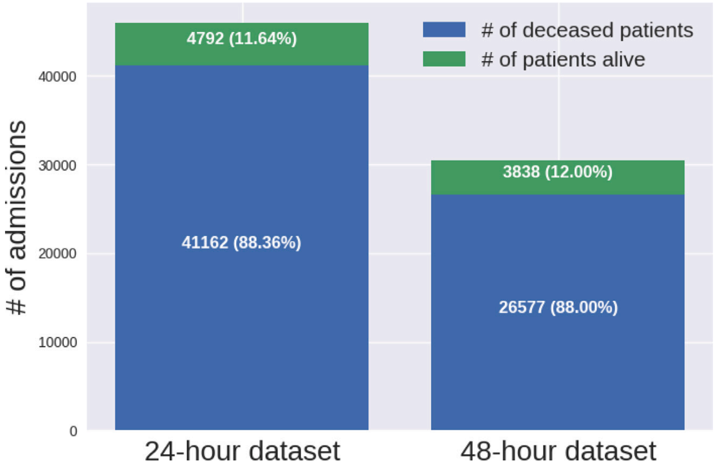


Fig. 5. Class distribution of the 24- and 48-hour datasets. Green squares represent number of admissions with deceased patients and blue squares represent number of admissions with living patients.

its performances were compared to normal binary cross-entropy and weighted binary cross-entropy. The focal loss is defined as follows:

$$\mathcal{L}_{fl} = -\frac{1}{N}\sum_{n=1}^{N} -w_{fl1}(1-\hat{y}_n)^{w_{fl2}} log(\hat{y}_n) \qquad \hat{y}_n = \begin{cases} p_n & \text{if } y_n = 1 \\ 1-p_n & \text{otherwise.} \end{cases} \tag{14}$$

where p_n is the predicted probability of the model for $y_n = 1$, $w_{fl1} \in [0,1]$ is a weighting factor used to balance the importance between them, and $w_{fl2} \geq 0$ is a focusing parameter applied to focus on the minority class. These are hyperparameters defined by the user.

The outcomes of the experimental evaluation are presented in the section that follows.

4. Experiment results and comparisons

To evaluate the performance of ALNN-GRU, we have chosen the task of predicting mortality in the ICU as a pilot case. Indeed, death is probably the worst of the adverse events that may occur during medical stays. Although paradoxically, the mortality rate in hospitals is higher than in Intensive Care Units (Capuzzo et al., 2014), the patients who enter intensive care are those whose prognosis is life-threatening. In such an emergency environment, where critical patient arrival rates can rapidly increase, decision-making can be very tricky, even for an experienced physician. As deep learning models have shown superior results compared to probabilistic and machine learning models on classification tasks, our model, which is a deep learning-based model, seems to be an ideal solution to support physicians' decision-making. Furthermore, as ICU data suffer from missing values due to time irregularity, these data are appropriate for evaluating the performance of the ALNN-GRU, which was originally built to handle irregular time series data while improving the accuracy of the downstream task.

In the following section, we describe the environment in which our model is run.

4.1. Settings

We coded the proposed model using the Python 3.0 programming language and the machine learning libraries Keras 2.4.3 and TensorFlow 2.4.0. All remaining pre-processing and performance evaluations were done with NumPy, Pandas, and Scikit-learn libraries. Finally, we ran the code on a computer cluster with the following characteristics: The AMD Threadripper 3960X is a 24-core, 48-thread processor with 128 GB of memory. It is paired with an NVidia 3090 GPU with 24 GB of graphics memory. The code is available at ALNN-GRU.

In the following, we thoroughly describe the dataset used in this experiment.

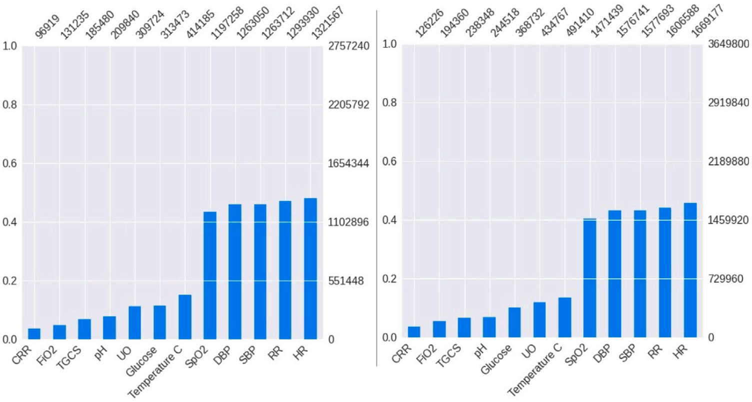
Table 3
Physiological measures.

Acronym	Definition	Type
SPO2	Oxygen Saturation	numeric
DBP	Diastolic Blood Pressure	numeric
SBP	Systolic Blood Pressure	numeric
UO	Urine Output	numeric
Temp	Temperature	numeric
FiO2	Fraction Inspired Oxygen	numeric
HR	Heart Rate	numeric
RR	Respiratory Rate	numeric
TGCS	Total Glasgow Coma Score	numeric
pH	pH	numeric
Glucose	Glucose	numeric
CRR	Peripheral Capillary Refill Oxygen Saturation	categorical

4.2. Dataset

We conducted our experiment with the publicly available database MIMIC-III, which contains anonymized health-related data from over forty thousand patients who stayed in the Beth Israel Deaconess Medical Center intensive care units between 2001 and 2012. From the chartevents and outputevents tables, we have extracted patient data from the first 24 hours (37,375 patients) and 48 hours (25,755 patients). As a patient may have several admissions, for the 24-hour dataset, we obtained 45,954 admissions records, 41,162 with label 0, and 4,792 (11.64%) with label 1. Whereas, for the 48 hour-dataset, we obtained 30,415 admissions records, 26,577 with the label 0 and 3,838 (12%) with the label 1. The class distribution of the two datasets is shown in Fig. 5. Inspired by Purushotham, Meng, Che, and Liu (2018), Shukla and Marlin (2019a), the physiological measures selected are described in Table 3.

In the next section, we describe the preprocessing steps performed before training the model.



24-hour dataset 48-hour dataset

Fig. 6. Missing value rate by feature: The left axis of each plot is the number of data points (observed and missing) per feature in percent. The top axis of each plot is the number of observed values per feature. The right axis of each plot is the number of data points (observed and missing) per feature.

4.3. Pre-processing

Before diving into the pre-processing details, it should be noted that the pre-processing we perform here is not intended for ad hoc time bin management but rather to deal with outliers (identified with practitioners' assistance), missing values, time series with different lengths, and worthless features. In fact, for certain medical or technical reasons, it may happen that during a patient visit, certain features from Table 3 are not recorded. This results in samples (multivariate time series) whose corresponding features have no value. The missing value rate for data collected in the first 24 and 48 hours is 74.57% and 76.53%, respectively. Fig. 6 shows the rate of missing values per feature in both datasets

The first step in pre-processing is to deal with outliers and missing values. We removed outliers with practitioners' assistance and considered them missing values. Missing values can occur at the beginning, middle, or/and end of a time series. If the time series contains missing values in the middle, we first interpolate them using the Pandas time interpolation method (*c.f.* Fig. 7, 1-preprocessing). The interpolation is based on the values surrounding the missing value and their respective timestamps. If a missing value appears at the beginning or end of the time series, we cannot use interpolation, as it relies on the surrounding values. Therefore, for that occurring at the beginning, we use the Pandas' backward fill method that relies on the value recorded after the current missing value (*c.f.* Fig. 7, 1-2-preprocessing). For the missing value occurring at the end of the time series, we use the Pandas' forward fill method that relies on the value recorded before the current missing value (*c.f.* Fig. 7, 1-2-preprocessing). The Pandas' forward fill method is also used for padding purposes. In other words, we use it to obtain univariate time series with the same number of observations. The last observed value and its corresponding timestamp are duplicated (*c.f.* Fig. 7, 2-preprocessing). Although the problem of time series with a number of observations is an underlying problem of temporal irregularity, the padding that we perform to remedy it does not solve the irregular temporal problem. We apply this because ALNN requires inputs with the same number of observations. To mitigate the impact of padded values in the model calculation, a binary mask is associated with them (*c.f.* Eq. (7)).

Unlike outliers and missing values, whose respective timestamps are recorded, we do not have timestamps for worthless features. Therefore, since ALNN requires timestamps, we need to impute both values and their respective timestamps. We set the timestamp values to 0, and we fill in worthless features with the empirical mean of the corresponding features (*c.f.* Fig. 7, 3-preprocessing). Since the impact of a value depends on its time lag penalty score (*c.f.* Section 3.2.1), the values with 0 as timestamp will be less involved in the model calculation. Additionally, a binary value is associated with each value to allow the model to rely less on imputed values (*c.f.* Eq. (7))

Although we do not perform any imputation/interpolation at the model's core that might be a noise factor, we do impute/interpolate values during processing that might generate noise. However, as Fig. 8 shows, the distribution of the original data is almost similar to that obtained after imputation. This observation ensures that the different imputation strategies we have used have resulted in minimal distortion of the original data structure. As a result, less noise is generated

In the next section, we present the properties and hyperparameters of the model.

4.4. Model properties and hyperparameters

Experiments were performed with values collected, respectively, at 24 and 48 hours after patient admission. The number of values observed for each patient's physiological measurements (features) during the admission may differ. Since the deep learning model requires inputs to have the same number of values, we define a standard length for each feature. After an extensive grid search, it was found that 60 was the optimal number of values per feature for the 24 hour-dataset. Regarding data loss, only 1.28% of features have more than 60 values. As we double the hours, we intuitively double the number of values per feature for the 48-hour dataset. Therefore, for the latter, the number of values per feature was set at 160. Concerning data loss, only 0.46% of features have more than 120 values. To link the mathematical notation in Section 3.1.1 to the experiment, for the 24−hour dataset X, T, M and $\Delta \in \mathbb{R}^{60\times 12}$. For the 48−hour dataset X, T, M and $\Delta \in \mathbb{R}^{120\times 12}$. 12 is the number of physiological measures (*c.f.* Table 3).

The reference time point vectors (*c.f.* Section 3.1.2) are set to $0-24$ and $0-48$ for the 24− and 48−hour datasets, respectively. Δr was set to 1 for both. Then, if we denote by $\mathbf{r}_{24}$ and $\mathbf{r}_{48}$ the set reference time points for the 24− and 48−hour datasets, respectively, we will have $\mathbf{r}_{24} = [0, 1, 2, 3, \dots, 24]$ and $\mathbf{r}_{48} = [0, 1, 2, 3, \dots, 48]$. Suppose we had set Δr to 0.5, then we would have had $\mathbf{r}_{24} = [0, 0.5, 1, 1.5, \dots, 24]$ and $\mathbf{r}_{48} = [0, 0.5, 1, 1.5, \dots, 48]$. The smaller Δr is, the closer we get to a set of continuous numbers, hence the claim that our function can approximate the behavior of a continuous function.

By a grid search procedure, we chose *relu* as the activation function at value-level extraction and feature-level aggregation (*c.f.* Section 3.2.2). More precisely, in Eqs. (7)–(10), the activation function

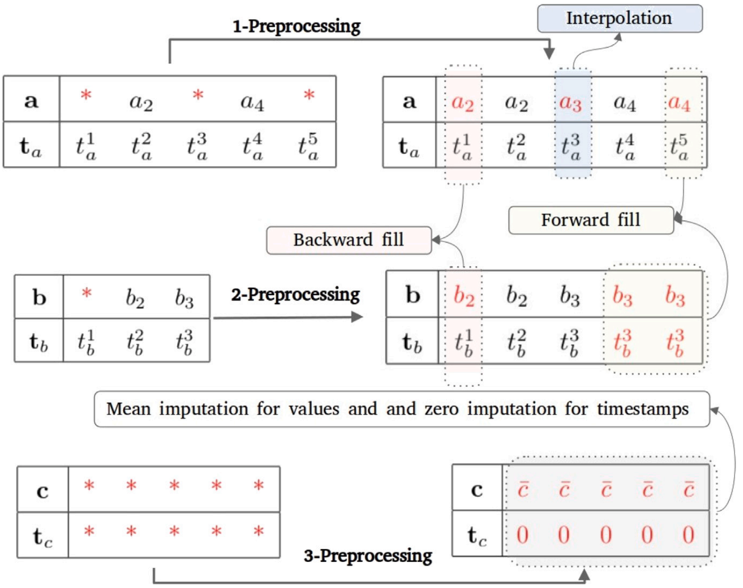


Fig. 7. Imputation workflow. **a**, **b** and **c** are three univariate time series belonging to the same sample with different missing value scenarios. * is the missing value indicator. Values in red are imputed values. $\bar{c}$ is the empirical mean of the feature **c**. After preprocessing, all univariate time series have the same number of observations and timestamps. It should be noted that there is no other step in the pre-processing. The numbering is used for referencing purposes only.

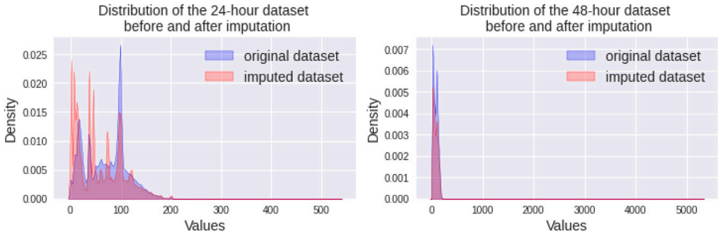


Fig. 8. Comparison of distributions of the two datasets before and after imputation of missing values.

$\sigma(.)$ used is *relu*. Due to the consideration of many parameters, we also apply a dropout=0.05 at these two levels to overcome the overfitting problem (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Then Eqs. (9) and (10) are reformulated as follows:

$$\mathbf{V} = Dropout(\sigma(\phi(<\mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I}> \odot \hat{W} + \hat{B})); 0.5) \tag{15}$$

$$Z = Dropout(\sigma(\phi(\bar{\mathbf{V}} \odot \bar{W}_z + \bar{B}_z)); 0.5) \tag{16}$$

The number of GRU units was set to 168 (*c.f.* Eq. (11)). Focal loss weighted factors were set to $w_{fl1} = 0.10$ and $w_{fl2} = 2.10$ for the previous 24 and 48 hours (*c.f.* Eq. (14)). The batch size value was set to 2000, and the number of epochs to 100. We use the Adam optimizer (Kingma & Ba, 2014) with a learning rate equal to 0.001.

As the k-fold cross-validation value is frequently set at 5 in several works (Che et al., 2018; Jun, Mulyadi, Choi, & Suk, 2020; Shukla & Marlin, 2019a), we also use the same value to train and test our model and those of our competitors. The training configuration and model's hyperparameters are summarized in Table 4. Apart from the number of epochs and GRU units, which differ from competing models, all parameters remain the same. After an extensive grid search, the number of epochs of the GRU-Δt, GRU-Δt-zero and BRITS-GRU-Δt was set at 50. For Interp-net (Neil et al., 2016), GRU-D, Bi-LSTM-Δt, mTAND (Shukla & Marlin, 2021a), ODE-LSTM (Lechner & Hasani, 2020), Phased-LSTM (Neil et al., 2016) and Neural-CDE (Chen et al., 2018), it was set at 100. The number of GRU units was set at 200 for all competitors integrating this layer. The Training configuration and hyperparameters of ALNN-GRU are summarized in Table 4.

Table 4
Training configuration and hyperparameters of ALNN-GRU.

Hyperparameters	Value	Hyperparameters	Value
GRU-units	168	ALNN activation functions	Relu
Δr	1	dropout at the values-level extraction	0.05
Batch size	2000	dropout at the features-level aggregation	0.05
epochs	100	Focal loss weighted factors w_{fl1}/w_{fl2}	0.1/2.1
Learning rate	0.001	k-fold cross-validation	5

4.5. Prediction performance

We usher in this section with a focus on mortality prediction accuracy. We compare the average Area under the ROC Curve (AUC), the Area Under the Precision-Recall Curve (AUPRC) scores and F1-scores

Table 5
Performances of the mortality prediction task (mean ±standard deviation from 5-cross validation).

Models	24			48		
	AUC	AUPRC	F1-score	AUC	AUPRC	F1-score
Bi-LSTM-Δt	0.817 ± 0.005	0.413 ± 0.012	0.367 ± 0.015	0.805 ± 0.017	0.434 ± 0.026	0.397 ± 0.023
BRITS-GRU-Δt	0.819 ± 0.006	0.406 ± 0.010	0.364 ± 0.003	0.805 ± 0.008	0.418 ± 0.020	0.405 ± 0.006
GRU-D	0.843 ± 0.006	0.448 ± 0.011	0.392 ± 0.008	0.826 ± 0.004	0.463 ± 0.019	0.422 ± 0.011
GRU-Δtzero	0.782 ± 0.012	0.334 ± 0.017	0.343 ± 0.009	0.756 ± 0.009	0.348 ± 0.022	0.358 ± 0.007
GRU-Δt	0.829 ± 0.009	0.430 ± 0.018	0.389 ± 0.008	0.818 ± 0.004	0.448 ± 0.013	0.419 ± 0.007
Interp-net	0.803 ± 0.005	0.359 ± 0.009	0.352 ± 0.003	0.754 ± 0.005	0.313 ± 0.005	0.359 ± 0.008
mTAND	0.840 ± 0.007	0.416 ± 0.010	0.388 ± 0.005	0.807 ± 0.004	0.442 ± 0.017	0.411 ± 0.006
Neural-CDE	0.798 ± 0.020	0.385 ± 0.017	0.362 ± 0.012	0.793 ± 0.005	0.401 ± 0.010	0.388 ± 0.012
ODE-LSTM	0.816 ± 0.009	0.396 ± 0.015	0.370 ± 0.011	0.806 ± 0.003	0.431 ± 0.016	0.410 ± 0.008
Phased-LSTM	0.841 ± 0.004	0.458 ± 0.016	0.394 ± 0.005	0.831 ± 0.004	0.483 ± 0.021	0.433 ± 0.012
ALNN-GRU	**0.854 ± 0.004**	**0.481 ± 0.013**	**0.400 ± 0.018**	**0.846 ± 0.004**	**0.507 ± 0.013**	**0.445 ± 0.013**

over the 5-fold cross-validation that we obtained against those of the state-of-the-art models shown below:

- Bi-LSTM-Δt: It is a bidirectional LSTM whose inputs are the concatenation of matrices of values, timestamps and masks;
- BRITS-GRU-Δt: It is a GRU-Δt whose missing values are imputed using the BRITS model (Cao et al., 2018b). BRITS is a bidirectional model that imputes missing values using recursive units and linear regression. BRITS and GRU-Δt are trained in end-to-end fashion;
- Interp-net (Shukla & Marlin, 2019a): Interpolation neural network that uses a set of Radial Basis Functions to interpolate missing values against a set of reference time points;
- GRU-D (Che et al., 2018): GRU-based model with a decay mechanism applied in its core which aims to impute missing values according to the duration of their absence;
- GRU-Δt: GRU whose inputs concatenate the observed values, their masks, and the time interval matrix. Irregularity and missing value patterns are implicitly captured through its inputs. No explicit mechanism is applied in its core;
- GRU-Δt-zero: GRU-Δt whose missing values are imputed with the value 0;
- mTAND (Shukla & Marlin, 2021a): It is an attention-RNN-based model consisting of an encoder and a decoder. The encoder is responsible for encoding a set of latent variables as a function of a set of reference time points and the observed value, along with their respective time stamps. Technically, the latent variables are sampled from a distribution learned via the output of multi-time attention and a set of reference time points. The decoder, on the other hand, interpolates the missing values into a set of requested time points;
- Neural-CDE (Chen et al., 2018): A neural network whose hidden layers at time t are calculated with an ordinary differential equation having as parameters the initial hidden state, the derivative of the function that calculates the hidden state, and the targeted timestamp (it can be a list of timestamps if multiple hidden states are needed);
- ODE-LSTM (Lechner & Hasani, 2020): LSTM-based model whose hidden states are calculated with ordinary differential equations;
- Phased-LSTM (Neil et al., 2016): LSTM-based model with an additional gate called the time gate. This gate is dedicated to a periodic update of the hidden and the memory cells of the LSTM according to a set of signals calculated in its different units.

In addition to the AUC, AUPRC and F1 scores, we also present in the Appendix B the confusion matrix for each model in the first cross-validation (i.e. the first loop of the cross-validation). Although, compared with the average of the AUC, AUPRC and F1 scores for the 5-fold cross-validation, the confusion matrix for the first cross-validation cannot accurately reflect the performance of the models, it gives us an overview of the Sensitivity and Specificity of each model.

On both datasets (24 h and 48 h), we can see in Table 5 that our model has a better AUC, AUPRC and F1-score. This confirms the effectiveness of the **hypotheses** we formulated in the 3.1.3 subsection and the model design choices. Among the competitors, we note that the Phased-LSTM and GRU-D models are the ones that perform the best. What may explain the effectiveness of GRU-D is the consideration of the property observed in human body health data (Vodovotz, An, & Androulakis, 2013) when imputing missing values. The effectiveness of Phased-LSTM, on the other hand, is due to its temporal gate which allows it to retain the original underlying temporal structure and handle temporal irregularities without requiring any imputation which could be a noise factor. Although the above models implement different practical approaches to dealing with temporal irregularity, the superior performance of our model highlights the fact that the duplication strategy driven by an exponential time decay mechanism deals better with irregular multivariate time series and makes the downstream task more accurate

Surprisingly, we find that GRU-Δt, which is the simplest model, performs better than more sophisticated models such as Interp-net, ODE-LSTM, Neural-CDE, BRITS-GRU-Δt and Bi-LSTM-Δt. This shows that, without a built-in explicit mechanism dedicated to solving the temporal irregularity problem, a simple GRU can effectively capture this irregularity through additional inputs (mask and time interval matrices) and, consequently, provide functional results. The difference between GRU-Δt and GRU-Δt-zero is that they implement different imputation techniques, namely mean imputation and zero imputation, respectively. Therefore, the superior performance of GRU-Δt indicates that mean imputation (which we also perform during the out-imputation[4]) should be preferred to zero imputation.

Furthermore, we note that compared to the Interp-net and mTAND models, which also perform an alignment process via a set of reference time points, our model is the most accurate. This can be explained by the fact that, in our model, the set of reference time points is strictly defined according to the data collection period (for example, if the data were collected for 10 hours, the minimum reference time point would be 0 or 1 and the maximum 10, see Subsection 3.1.2). This is not the case with the Interp-net and mTAND models, where the only constraint is to have a regular time interval between reference points. As a result, the underlying temporal structure associated with the new values (which can be latent) calculated at these reference time points may be considerably lost.

Another interesting analysis is the variation in the AUC, AUPRC and F1 scores of the models. From Table 6, we observed that when more data are considered (48-hour dataset), the AUC score of all models decreases and their AUPRC (except Interp-net) and F1 scores increase (despite a higher percentage of missing values). The higher percentage of the targeted class in the 48-hour-dataset justifies this variation. For the AUC score, our model achieves the second-lowest decay percentage.

---

[4] Imputation performed during the preprocessing stage.

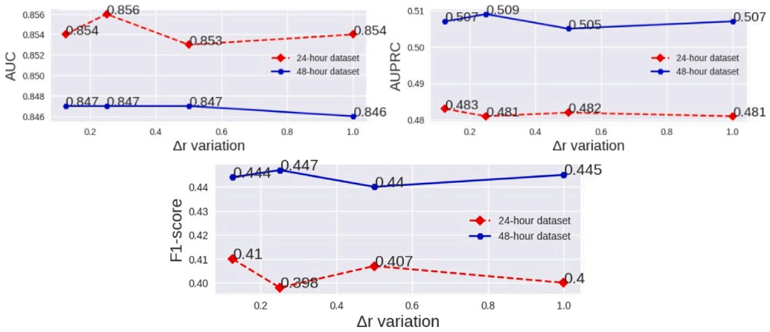


Fig. 9. AUC, AUPRC and F1 scores by varying Δr for the 24- and 48-hour datasets. As we can notice, there is no linear correlation between Δr and the metric scores.

Table 6
AUC, AUPRC and F1-score variations from 24 to 48 h data.

Models	AUC variation ↘	AUPRC variation ↗	F1-score variation ↗
Bi-LSTM-Δt	0.012	0.021	0.030
BRITS-GRU-Δt	0.014	0.012	0.041
GRU-D	0.017	0.015	0.030
GRU-Δrzero	0.026	0.014	0.015
GRU-Δt	0.011	0.018	0.030
Interp-net	0.049	0.046 ↘	0.007
mTAND	0.033	0.026	0.023
Neural-CDE	**0.005**	0.016	0.026
ODE-LSTM	0.010	**0.035**	0.040
Phased-LSTM	0.010	0.025	0.039
ALNN-GRU	**0.008**	**0.027**	**0.045**

On the other hand, for the AUPRC and F1 scores, our model achieves the second and first-best growth percentages, respectively. From this analysis, if we have to compromise between the AUC score, the AUPRC score, the F1-score and their respective variations, our model turns out to be the perfect model to consider when more data with a more significant number of the targeted class is available.

4.6. Ablation studies

In what follows, we put the focus on the outcomes of the following ablation studies:

1. We compare our approach to the classical approach, which consists of discretizing time;
2. We train our model with two other loss functions and compare the results;
3. We remove the ALNN or the GRU to assess their respective contribution;
4. We apply two other imputation strategies and compare them to the one used and
5. We evaluate the performance of our model with different values of Δr and show how this parameter affects AUC and AUPRC scores.

4.6.1. GRU-mask (time discretized) versus ALNN-GRU

To feed the GRU, we collect data every 1 hour. We then obtain a regular multivariate time series. We performed a mean imputation to fill in the hour bin with no value. The average is considered if many observations are collected in an hour bin. Additionally, a mask matrix indicating which value was imputed is concatenated with the multivariate time series. We call this GRU variant the GRU-mask. Table 7 sheds light on the cruciality of considering all the observed values and transforming them into pseudo-aligned latent values via the ALNN before feeding them into a GRU for the mortality prediction task.

4.6.2. Alternative loss functions

To check whether the focal loss is the most suitable to address the unbalanced data problem, we evaluate our model with two other loss functions: the Binary cross Entropy (17) and the Weighted Binary cross Entropy (18).

$$\mathcal{L}_{bc} = -\frac{1}{N}\sum_{n=1}^{N}[y_n * \ln(\hat{y}_n) + (1 - y_n) * \ln(1 - \hat{y}_n)] \tag{17}$$

$$\mathcal{L}_{wbc} = -\frac{1}{N}\sum_{n=1}^{N}[w_1 * y_n * \ln(\hat{y}_n) + w_0 * (1 - y_n) * \ln(1 - \hat{y}_n)] \tag{18}$$

$w_0 = N/(2 * N_0)$ and $w_1 = N/(2 * N_1)$, where N is the number of samples, N_0 is the number of samples of the class 0, and N_1 is the number of samples of the class 1. Indeed, w_1 makes it possible to penalize the model more whenever class 1 is misclassified, whereas w_0 makes it possible to penalize the model less whenever class 0 is misclassified.

In Table 8, we notice that the focal loss addresses the unbalanced data problem much better than other loss functions.

4.6.3. Contribution of ALNN versus GRU

Since the output of the ALNN is in matrix form, we add a Linear layer to it, followed by a Dense layer with a sigmoid activation function. We call this combination ALNN-FNN, where FNN stands for Feedforward Neural Network. The GRU here is actually a GRU-Δt since its inputs are the matrix of values, time intervals, and masks. In Table 9, the higher scores of the ALNN-FNN compared to those of the GRU-Δt highlight the fact that ALNN contributes more than GRU when combined. However, the performances of ALNN-FNN are lower than those of ALNN-GRU. Indeed, a linear layer cannot capture temporal patterns as well as the GRU does

4.6.4. Different imputation strategies

Zero and median imputation are simple imputation strategies commonly used in the literature. We then compare them to the one we used. The results in Table 10 show that the combination of interpolation, empirical mean, and backward and forward fill is much more effective than the simple zero and median imputation strategies

4.6.5. Study impact of the Δr variation

In this study, we varied the value of this parameter in $\{0.125, 0.25, 0.5, 1\}$. From Fig. 9, we notice no linear correlation between Δr and metric scores. In some cases, like the one with $\Delta r = 0.25$ for the 24 hour-dataset, we obtain a better AUC score than the one obtained with the default value, i.e., $\Delta r = 1$. In addition, with $\Delta r = 0.5$ for the 48 hour-dataset, the AUPRC score is lower than that obtained with the default value. On the one hand, adding more reference time points by decreasing the Δr value does not guarantee a better score. Indeed,

Table 7
Performance of the GRU (time discretized) vs. ALNN-GRU.

Models	24			48		
	AUC	AUPRC	F1-score	AUC	AUPRC	F1-score
GRU-mask	0.832 ± 0.008	0.439 ± 0.019	0.388 ± 0.004	0.824 ± 0.004	0.469 ± 0.017	0.422 ± 0.007
ALNN-GRU	**0.854 ± 0.004**	**0.481 ± 0.013**	**0.400 ± 0.018**	**0.846 ± 0.004**	**0.507 ± 0.013**	**0.445 ± 0.013**

Table 8
Evaluation of loss function impact.

Prior hours	Loss function	AUC	AUPRC	F1-score
24	$\mathcal{L}_{bc}$	0.843 ± 0.004	0.451 ± 0.014	0.398 ± 0.006
	$\mathcal{L}_{wbc}$	0.842 ± 0.003	0.453 ± 0.014	0.396 ± 0.008
	$\mathcal{L}_{fl}$ **(ours)**	**0.854 ± 0.004**	**0.481 ± 0.013**	**0.400 ± 0.018**
48	$\mathcal{L}_{bc}$	0.842 ± 0.008	0.489 ± 0.020	0.440 ± 0.009
	$\mathcal{L}_{wbc}$	0.841 ± 0.006	0.489 ± 0.022	0.439 ± 0.011
	$\mathcal{L}_{fl}$ **(ours)**	**0.846 ± 0.004**	**0.507 ± 0.013**	**0.445 ± 0.013**

Table 9
ALNN-FNN versus ALNN-GRU versus GRU-Δt.

Models	24			48		
	AUC	AUPRC	F1-score	AUC	AUPRC	F1-score
GRU-Δt	0.829 ± 0.009	0.430 ± 0.018	0.389 ± 0.008	0.818 ± 0.004	0.448 ± 0.013	0.419 ± 0.007
ALNN-FNN	0.847 ± 0.005	0.464 ± 0.021	0.396 ± 0.021	0.844 ± 0.006	0.490 ± 0.015	0.432 ± 0.007
ALNN-GRU	**0.854 ± 0.004**	**0.481 ± 0.013**	**0.400 ± 0.018**	**0.846 ± 0.004**	**0.507 ± 0.013**	**0.445 ± 0.013**

Table 10
ALNN-GRU performances with different imputation strategies.

Imputation strategy	24			48		
	AUC	AUPRC	F1-score	AUC	AUPRC	F1-score
Median	0.838 ± 0.005	0.422 ± 0.110	0.388 ± 0.020	0.833 ± 0.002	0.446 ± 0.017	0.405 ± 0.012
Zero	0.839 ± 0.006	0.412 ± 0.019	0.380 ± 0.014	0.825 ± 0.006	0.436 ± 0.013	0.400 ± 0.010
ALNN-GRU	**0.854 ± 0.004**	**0.481 ± 0.013**	**0.400 ± 0.018**	**0.846 ± 0.004**	**0.507 ± 0.013**	**0.445 ± 0.013**

if several reference time points do not match the input timestamps, the calculated latent values concerning these reference instants might introduce noise. On the other hand, increasing the Δr value to have fewer reference time points might result in a low impact of observed values with timestamps that do not match these reference time points. We do not currently find a theoretical formula that allows us to choose the value of Δr better. As with other hyperparameters, a grid search or similar techniques must be applied to find the most suitable one

5. Wrapping-up discussion

Although our proposal provides satisfactory results, it still has a lot of room for improvement. Indeed, the out-imputation and out-interpolation that we have performed to solve the problems of outliers, streams of different lengths, and worthless streams are based on a strong assumption that does not consider the objective function of the downstream task. In other words, they are not data-driven imputation or interpolation methods. Therefore, even though our duplication approach is a lower noise driver, we still inject noise during preprocessing. A straightforward technique that could be implemented to solve this problem would be to incorporate a mask layer that allows missing values (usually defined with a value of 0) to be skipped during any calculation performed in the model. The downside of this approach is that it discards the information carried by missing values, which can be crucial for a model dedicated to a sensitive task such as the prediction of mortality. The authors in Rubin (1976) stated that, usually, missing values carry relevant information that might help to discriminate the different classes involved in a classification problem. A better alternative to addressing this missing value problem could be using a generative model such as the Autoencoder. Integrating an Autoencoder makes it possible to steer the imputation process toward the objective function of the downstream task. In this way, we do not have to make strong assumptions about missing values. With an autoencoder, imputation can also take advantage of correlation within and between streams. This will be a plus, as we do not consider this aspect in our current work. If integrating the autoencoder in addition to our proposal seems like an efficient approach to handling missing values, certain aspects, such as the uncertainty of imputed values and the heteroscedasticity (in case we decide to use a Variational Autoencoder, (Kingma & Welling, 2013)), should be strongly considered.

6. Conclusion

We built a neural network-based model on top of an RNN model (GRU) to transform irregular multivariate time series data into *pseudo-aligned latent values*. As an RNN is composed of a series of discrete hidden layers, it is more suitable for processing regular sequences of observations. This property makes it a poor candidate when dealing with irregular time series. The ALNN is a preprocessing step that transforms irregular multivariate time series into a *pseudo-aligned latent* *values* to keep this property valid. Thanks to a duplication process driven by an exponential decay mechanism, the ALNN's core does not perform any imputation or interpolation that might be very noisy. The results show that the ALNN-GRU outperforms the state-of-the-art models in performance. Also, because you can change how long it takes between reference points, ALNN-GRU can behave in a way that is similar to a continuous function. As a result, it is an excellent candidate for dealing with time series, which are frequently defined in continuous time space. However, in return, a massive amount of calculation memory is required.

The fact that the output of the ALNN (the *pseudo-aligned latent values*) is in matrix form gives us a wealth of possibilities to improve the learning process, try different architectures such as combining a CNN with an RNN, and implement an explainable component, which is a must for a medical model. Regarding improving the learning process, in our future work (in addition to using Autoencoder to impute missing values), we first plan to calculate the correlation matrix from the *pseudo-aligned latent values* and integrate it efficiently into the model. Secondly, this correlation matrix with the *pseudo-aligned latent values* will be used to build an explainable component dedicated to highlighting physiological measures that may be the cause of death. Even though we conducted this work for a specific medical task, our proposal can be used for any classification or prediction task requiring irregular time series data as input. For this reason, we also want to see how well our model works on non-medical tasks requiring irregular time series data.

CRediT authorship contribution statement

Nzamba Bignoumba: Conceptualization, Methodology, Software, Validation, Visualization, Writing – review & editing. **Nedra Mellouli:** Writing – review & editing. **Sadok Ben Yahia:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgment

Our work has been partially conducted in the project "ICT programme", which was supported by the European Union through the European Social Fund. It is also supported by the EU-Astra - TUT Development Plan for 2016–2022 (ASTRA) reg no.2014–2020.4.1.16-0032.

Appendix A. Matrix representation of alnn formulas

A.1. Time lag penalty

$$I_p = X \odot exp\{-max(0, -\gamma_p|r_p - T|)\}; I_p \in \mathbb{R}^{J\times K} \tag{A.1}$$

$$\mathbf{I} = \begin{bmatrix} I_1, & \cdots & , I_P \end{bmatrix}^{\mathsf{T}}; \mathbf{I} \in \mathbb{R}^{N'\times Q} \tag{A.2}$$

$N' = 1$ and $Q = P \times J \times K$.

A.2. Alignment

$$\mathbf{X} = \underbrace{[X, \dots, X]}_{1,\dots,P}^{\mathsf{T}},\ \mathbf{M} = \underbrace{[M, \dots, M]}_{1,\dots,P}^{\mathsf{T}},\ \boldsymbol{\Delta} = \underbrace{[\Delta, \dots, \Delta]}_{1,\dots,P}^{\mathsf{T}} \tag{A.3}$$

$$< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} >= \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} x_1^1 & m_1^1 & \delta_1^1 & (i_1^1)^1 \\ & \vdots & & \\ x_1^K & m_1^K & \delta_1^K & (i_1^K)^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_J^1 & m_J^1 & \delta_J^1 & (i_J^1)^1 \\ & \vdots & & \\ x_J^K & m_J^K & \delta_J^K & (i_J^K)^1 \end{bmatrix} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \begin{bmatrix} x_1^1 & m_1^1 & \delta_1^1 & (i_1^1)^P \\ & \vdots & & \\ x_1^K & m_1^K & \delta_1^K & (i_1^K)^P \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_J^1 & m_J^1 & \delta_J^1 & (i_J^1)^P \\ & \vdots & & \\ x_J^K & m_J^K & \delta_J^K & (i_J^K)^P \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{A.4}$$

$$< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} >\in \mathbb{R}^{N'\times Q\times 4} \tag{A.5}$$

A.2.1. Values-level extraction

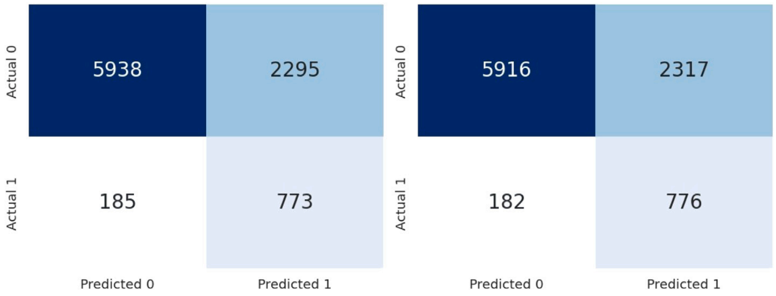
$$(v_j^k)_p = \sigma(x_j^k(\hat{w}_1^k)_j^p + m_j^k(\hat{w}_2^k)_j^p + \delta_j^k(\hat{w}_3^k)_j^p + (i_j^k)_p(\hat{w}_4^k)_j^p + (\hat{b}^k)_j^p) \tag{A.6}$$

$\sigma(.)$ is an activation function. $\hat{W}((\hat{w}_{[1,4]}^k)_j^p) \in \mathbb{R}^{1\times Q\times 4}$ and $\hat{B}((\hat{b}^k)_j^p) \in \mathbb{R}^{1\times Q\times 1}$ are learnable parameters.

$$\mathbf{V} = \sigma(\phi(< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} > \odot\hat{W} + \hat{B}));\ (v_j^k)_p \in \mathbf{V} \in \mathbb{R}^{N'\times Q\times 1} \tag{A.7}$$

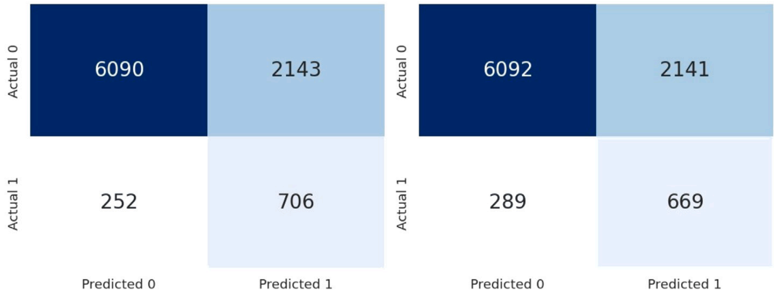
where $\phi(.)$ is the sum function of the coefficients of $(< \mathbf{X}, \mathbf{M}, \boldsymbol{\Delta}, \mathbf{I} > \odot\hat{W} + \hat{B}) \in \mathbb{R}^{N'\times Q\times 4}$ along the last axis (see Eq. (A.6)). $\odot$ is the Hadamard product.

$$\hat{W} = \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} (\hat{w}_1^1)_1^1 & (\hat{w}_2^1)_1^1 & (\hat{w}_3^1)_1^1 & (\hat{w}_4^1)_1^1 \\ & \vdots & & \\ (\hat{w}_1^K)_1^1 & (\hat{w}_2^K)_1^1 & (\hat{w}_3^K)_1^1 & (\hat{w}_4^K)_1^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\hat{w}_1^1)_J^1 & (\hat{w}_2^1)_J^1 & (\hat{w}_3^1)_J^1 & (\hat{w}_4^1)_J^1 \\ & \vdots & & \\ (\hat{w}_1^K)_J^1 & (\hat{w}_2^K)_J^1 & (\hat{w}_3^K)_J^1 & (\hat{w}_4^K)_J^1 \end{bmatrix} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \begin{bmatrix} (\hat{w}_1^1)_1^P & (\hat{w}_2^1)_1^P & (\hat{w}_3^1)_1^P & (\hat{w}_4^1)_1^P \\ & \vdots & & \\ (\hat{w}_1^K)_1^P & (\hat{w}_2^K)_1^P & (\hat{w}_3^K)_1^P & (\hat{w}_4^K)_1^P \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\hat{w}_1^1)_J^P & (\hat{w}_2^1)_J^P & (\hat{w}_3^1)_J^P & (\hat{w}_4^1)_J^P \\ & \vdots & & \\ (\hat{w}_1^K)_J^P & (\hat{w}_2^K)_J^P & (\hat{w}_3^K)_J^P & (\hat{w}_4^K)_J^P \end{bmatrix} \end{bmatrix} \end{bmatrix}, \hat{B} = \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} (\hat{b}^1)_1^1 \\ \vdots \\ (\hat{b}^K)_1^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\hat{b}^1)_J^1 \\ \vdots \\ (\hat{b}^K)_J^1 \end{bmatrix} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \begin{bmatrix} (\hat{b}^1)_1^P \\ \vdots \\ (\hat{b}^K)_1^P \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\hat{b}^1)_J^P \\ \vdots \\ (\hat{b}^K)_J^P \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{A.8}$$



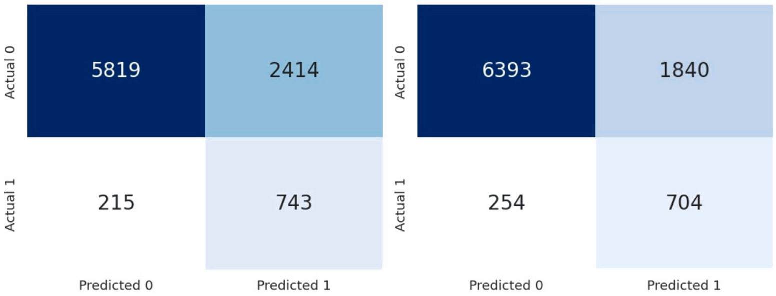
Phased-LSTM
- Specificity= 0.721
- Sensitivity= 0.807

GRU-D
- Specificity= 0.719
- Sensitivity= 0.810



Interp-net
- Specificity= 0.740
- Sensitivity= 0.737

GRU-Δt-zero
- Specificity= 0.740
- Sensitivity= 0.698



Bi-LSTM-Δt
- Specificity= 0.707
- Sensitivity= 0.776

GRU-Δt
- Specificity= 0.777
- Sensitivity= 0.735

Fig. A.10. Phased-LSTM, GRU-D, Interp-net, GRU-Δt-zero, Bi-LSTM-Δt and GRU-Δt confusion matrices for the 24-hour dataset.

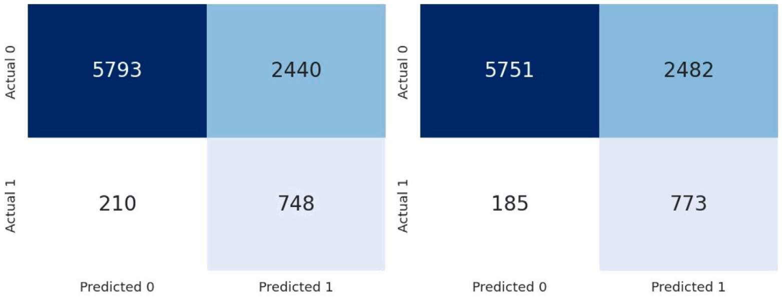
$$\mathbf{V} = \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} (v_1^1)^1 \\ \vdots \\ (v_1^K)^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (v_J^1)^1 \\ \vdots \\ (v_J^K)^1 \end{bmatrix} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \begin{bmatrix} (v_1^1)^P \\ \vdots \\ (v_1^K)^P \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (v_J^1)^P \\ \vdots \\ (v_J^K)^P \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{A.9}$$

A.2.2. Features-level aggregation

$$Z = \sigma(\phi(\bar{\mathbf{V}} \odot \bar{W}_z + \bar{B}_z)); Z \in \mathbb{R}^{P\times K} \tag{A.10}$$

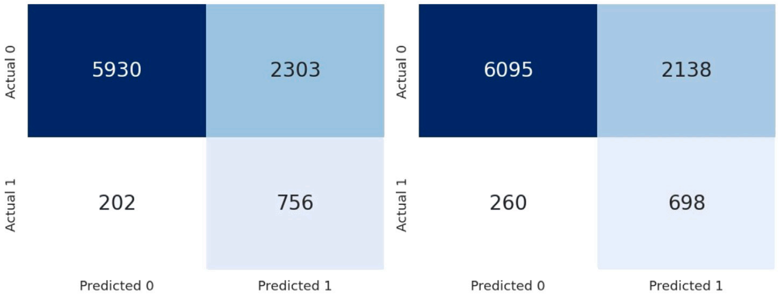
where $\sigma(.)$ is an activation function; $\phi(.)$ the sum function of the coefficients of $(\bar{\mathbf{V}} \odot \bar{W}_z + \bar{B}_z) \in \mathbb{R}^{N'\times Q}$ along the third axis ($N' = 1$ and $Q = P\times J\times K$). $\bar{V}(\bar{v}_j^k)_p \in \mathbb{R}^{N'\times Q}$ is the reshaped version of $V \in \mathbb{R}^{N'\times Q\times 1}$. $\bar{W}_z((\bar{w}_j^k)^p) \in \mathbb{R}^{1\times Q}$ and $\bar{B}_z((\bar{b}^k)^p) \in \mathbb{R}^{1\times P\times K}$ are model parameters.

$$Reshape(\mathbf{V}) = \bar{\mathbf{V}} = \begin{bmatrix} \begin{bmatrix} (v_1^1)^1 & \cdots & (v_1^K)^1 \\ & \vdots & \\ (v_J^1)^1 & \cdots & (v_J^K)^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (v_1^1)^P & \cdots & (v_1^K)^P \\ & \vdots & \\ (v_J^1)^P & \cdots & (v_J^K)^P \end{bmatrix} \end{bmatrix} \tag{A.11}$$



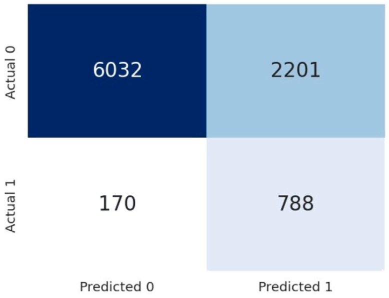
BRITS-LSTM-Δt
- Specificity= 0.704
- Sensitivity= 0.781

ODE-LSTM
- Specificity= 0.699
- Sensitivity= 0.807



mTAND
- Specificity= 0.704
- Sensitivity= 0.781

Neural-CDE
- Specificity= 0.740
- Sensitivity= 0.729



ALNN-GRU
- Specificity= 0.733
- Sensitivity= 0.823

Fig. A.11. BRITS-LSTM-Δt, ODE-LSTM, mTAND, Neural-CDE and ALNN-GRU confusion matrices for the 24-hour dataset.

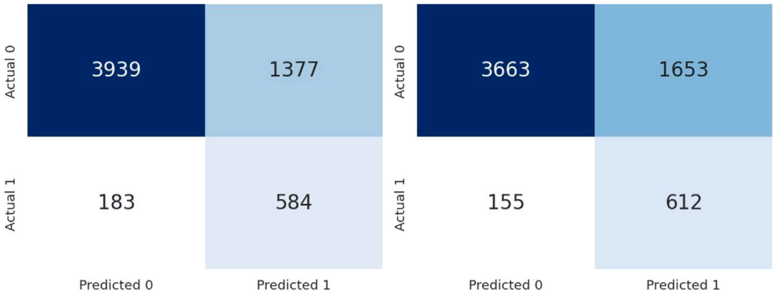
$$\bar{W}_z = \begin{bmatrix} \begin{bmatrix} (\bar{w}_1^1)^1 & \cdots & (\bar{w}_1^K)^1 \\ & \vdots & \\ (\bar{w}_J^1)^1 & \cdots & (\bar{w}_J^K)^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\bar{w}_1^1)^P & \cdots & (\bar{w}_1^K)^P \\ & \vdots & \\ (\bar{w}_J^1)^P & \cdots & (\bar{w}_J^K)^P \end{bmatrix} \end{bmatrix}, \bar{B}_z = \begin{bmatrix} \begin{bmatrix} (\bar{b}^1)^1 & \cdots & (\bar{b}^K)^1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\bar{b}^1)^P & \cdots & (\bar{b}^K)^P \end{bmatrix} \end{bmatrix} \tag{A.12}$$

$$Z = \begin{bmatrix} z_1^1 & \cdots & z_1^K \\ \vdots & \vdots & \vdots \\ z_P^1 & \cdots & z_P^K \end{bmatrix} \quad \begin{matrix} --> r_1 \\ \vdots \\ --> r_P \end{matrix} \tag{A.13}$$

$$z_p^k = \sigma(\sum_{i=1}^{J} (\bar{v}_i^k)_p (\bar{w}_i^k)^p + (\bar{b}^k)^p); z_p^k \in Z \tag{A.14}$$

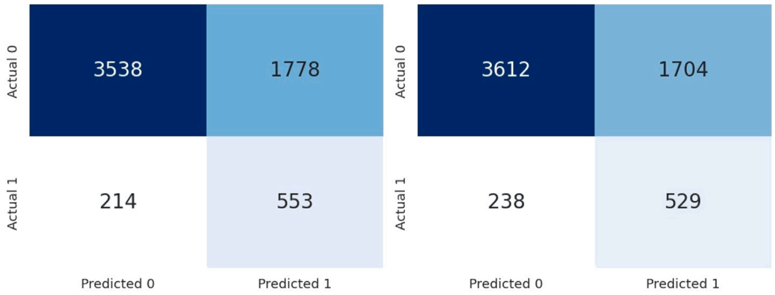
Appendix B. Confusion matrices

In this section, we present the confusion matrices of each model obtained during the first cross-validation of the 24-hour (Figs. A.10 and A.11) and 48-hour datasets (Figs. A.12 and A.13). Although, compared with the average of the AUC, AUPRC and F1 scores for the 5-fold cross-validation, the confusion matrix for the first cross-validation cannot accurately reflect the performance of the models, it gives us insight into the Sensitivity and the Specificity of each model. The results obtained show that, on the two datasets, our model can achieve a Specificity and



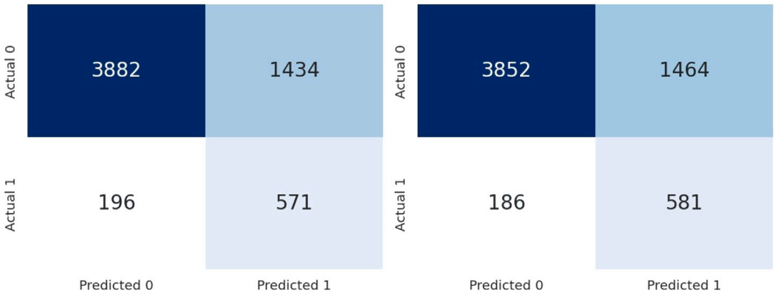
Phased-LSTM
- Specificity= 0.741
- Sensitivity= 0.761

GRU-D
- Specificity= 0.689
- Sensitivity= 0.798



Interp-net
- Specificity= 0.666
- Sensitivity= 0.721

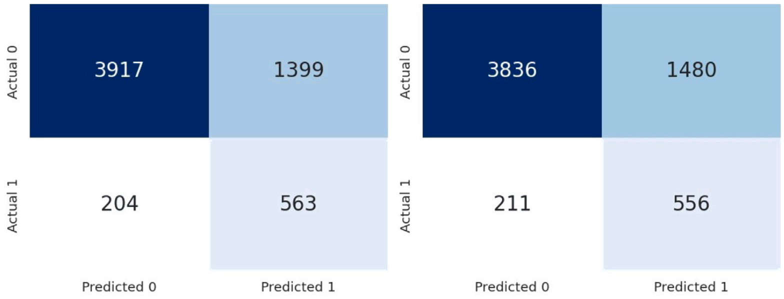
GRU-Δt-zero
- Specificity= 0.679
- Sensitivity= 0.690



Bi-LSTM-Δt
- Specificity= 0.730
- Sensitivity= 0.744

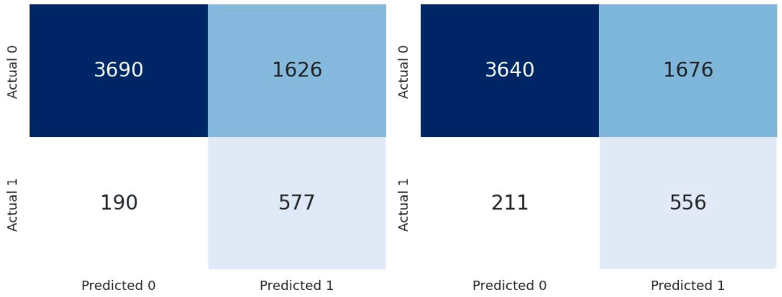
GRU-Δt
- Specificity= 0.725
- Sensitivity= 0.757

Fig. A.12. Phased-LSTM, GRU-D, Interp-net, GRU-Δt-zero, Bi-LSTM-Δt and GRU-Δt confusion matrices for the 48-hour dataset.



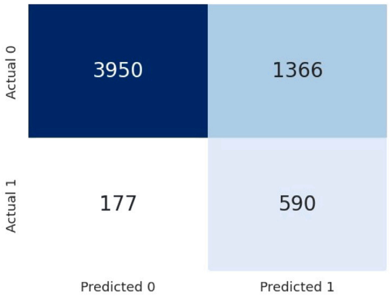
BRITS-LSTM-Δt
- Specificity= 0.737
- Sensitivity= 0.734

ODE-LSTM
- Specificity= 0.722
- Sensitivity= 0.725



mTAND
- Specificity= 0.694
- Sensitivity= 0.752

Neural-CDE
- Specificity= 0.685
- Sensitivity= 0.725



ALNN-GRU
- Specificity= 0.743
- Sensitivity= 0.769

Fig. A.13. BRITS-LSTM-Δt, ODE-LSTM, mTAND, Neural-CDE and ALNN-GRU confusion matrices for the 48-hour dataset.

Sensitivity score of around 70% and 80% respectively (see Figs. A.11 and A.13).

References

Binkowski, M., Marti, G., & Donnat, P. (2018). Autoregressive convolutional neural networks for asynchronous time series. In *International conference on machine learning* (pp. 580–589). PMLR.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018a). BRITS: bidirectional recurrent imputation for time series. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018* (pp. 6776–6786). URL https://proceedings.neurips.cc/paper/2018/hash/734e6bfcd358e25ac1db0a4241b95651-Abstract.html.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018b). Brits: Bidirectional recurrent imputation for time series. *Advances in Neural Information Processing Systems*, *31*.

Capuzzo, M., Volta, C. A., Tassinati, T., Moreno, R. P., Valentin, A., Guidet, B., et al. (2014). Hospital mortality of adults admitted to intensive care units in hospitals with and without intermediate care units: a multicentre European cohort study. *Critical Care*, *18*(5), 1–15.

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, *8*(1), 1–12.

Chen, Y., Ding, F., & Zhai, L. (2022). Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction. *Expert Systems with Applications*, *200*, Article 117011. http://dx.doi.org/10.1016/j.eswa.2022.117011.

Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, *31*.

Chen, J.-F., Wang, W.-M., & Huang, C.-M. (1995). Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. *Electric Power Systems Research*, *34*(3), 187–196.

Choi, E., Xiao, C., Stewart, W. F., & Sun, J. (2018). Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In S. Bengio, H. M. Wallach,

H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018* (pp. 4552–4562). URL https://proceedings.neurips.cc/paper/2018/hash/934b535800b1cba8f96a5d72f72f1611-Abstract.html.

Cini, A., Marisca, I., & Alippi, C. (2021). Filling the g_ap_s: Multivariate time series imputation by graph neural networks. arXiv preprint arXiv:2108.00298.

Dey, R., & Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In *IEEE 60th international midwest symposium on circuits and systems* (pp. 1597–1600). IEEE, http://dx.doi.org/10.1109/MWSCAS.2017.8053243.

Du, W., Côté, D., & Liu, Y. (2023). SAITS: self-attention-based imputation for time series. *Expert Systems with Applications*, *219*, Article 119619. http://dx.doi.org/10.1016/j.eswa.2023.119619.

El-Rashidy, N., El-Sappagh, S. H. A., AbuHmed, T., Abdelrazek, S., & El-Bakry, H. M. (2020). Intensive care unit mortality prediction: An improved patient-specific stacking ensemble model. *IEEE Access*, *8*, 133541–133564. http://dx.doi.org/10.1109/ACCESS.2020.3010556.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*(1), 1–9.

Jun, E., Mulyadi, A. W., Choi, J., & Suk, H.-I. (2020). Uncertainty-gated stochastic sequential model for ehr mortality prediction. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(9), 4052–4062.

Jun, E., Mulyadi, A. W., Choi, J., & Suk, H. (2021). Uncertainty-gated stochastic sequential model for EHR mortality prediction. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(9), 4052–4062. http://dx.doi.org/10.1109/TNNLS.2020.3016670.

Kidger, P., Morrill, J., Foster, J., & Lyons, T. (2020). Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, *33*, 6696–6707.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 1003–1012). IEEE Computer Society, http://dx.doi.org/10.1109/CVPR.2017.113.

Lechner, M., & Hasani, R. (2020). Learning long-term dependencies in irregularly-sampled time series. arXiv preprint arXiv:2006.04418.

Lee, Y., Jun, E., & Suk, H.-I. (2021). Multi-view integration learning for irregularly-sampled clinical time series. arXiv preprint arXiv:2101.09986.

Li, Z., Yu, J., Zhang, G., & Xu, L. (2023). Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting. *Expert Systems with Applications*, *216*, Article 119374. http://dx.doi.org/10.1016/j.eswa.2022.119374.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Lipton, Z. C., Kale, D., & Wetzel, R. (2016). Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine learning for healthcare conference* (pp. 253–270). PMLR.

MacKay, D. J. (1992). Bayesian interpolation. *Neural Computation*, *4*(3), 415–447.

Martínez, F., Frías, M. P., Pérez, M. D., & Rivera, A. J. (2019). A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, *52*(3).

Narayan Shukla, S., & Marlin, B. M. (2021). Multi-time attention networks for irregularly sampled time series. arXiv e-prints, arXiv–2101.

Neil, D., Pfeiffer, M., & Liu, S.-C. (2016). Phased lstm: Accelerating recurrent network training for long or event-based sequences. *Advances in Neural Information Processing Systems*, *29*.

Oskarsson, J., Sidén, P., & Lindsten, F. (2023). Temporal graph neural networks for irregular data. http://dx.doi.org/10.48550/arXiv.2302.08415, CoRR abs/2302.08415 arXiv:2302.08415.

Park, J., Artin, M. G., Lee, K. E., Pumpalova, Y. S., Ingram, M. A., May, B. L., et al. (2022). Deep learning on time series laboratory test results from electronic health records for early detection of pancreatic cancer. *Journal of Biomedical Informatics*, *131*, Article 104095. http://dx.doi.org/10.1016/j.jbi.2022.104095.

Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2018). Benchmarking deep learning models on large healthcare datasets. *Journal of Biomedical Informatics*, *83*, 112–134.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., & Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A (Mathematical and Physical Sciences)*, *371*(1984), Article 20110550.

Rubanova, Y., Chen, R. T., & Duvenaud, D. (2019). Latent ODEs for irregularly-sampled time series. arXiv Search in.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Shan, S., Li, Y., & Oliva, J. B. (2021). Nrtsi: Non-recurrent time series imputation. arXiv preprint arXiv:2102.03340.

Shukla, S. N., & Marlin, B. M. (2018). Modeling irregularly sampled clinical time series. arXiv arXiv:1812.00531.

Shukla, S. N., & Marlin, B. M. (2019a). Interpolation-prediction networks for irregularly sampled time series. arXiv preprint arXiv:1909.07782.

Shukla, S. N., & Marlin, B. M. (2019b). Interpolation-prediction networks for irregularly sampled time series. In *7th international conference on learning representations*. OpenReview.net, URL https://openreview.net/forum?id=r1efr3C9Ym.

Shukla, S. N., & Marlin, B. M. (2021a). Multi-time attention networks for irregularly sampled time series. arXiv preprint arXiv:2101.10318.

Shukla, S. N., & Marlin, B. M. (2021b). Heteroscedastic temporal variational autoencoder for irregularly sampled time series. arXiv preprint arXiv:2107.11350.

Song, H., Rajan, D., Thiagarajan, J. J., & Spanias, A. (2018). Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-second AAAI conference on artificial intelligence*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Suo, Q., Zhong, W., Xun, G., Sun, J., Chen, C., & Zhang, A. (2020). GLIMA: Global and local time series imputation with multi-directional attention learning. In *2020 IEEE international conference on big data* (pp. 798–807). IEEE.

Tan, Q., Ye, M., Ma, A. J., Yang, B., Yip, T. C., Wong, G. L., et al. (2021). Explainable uncertainty-aware convolutional recurrent neural network for irregular medical time series. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(10), 4665–4679. http://dx.doi.org/10.1109/TNNLS.2020.3025813.

Tan, Q., Ye, M., Yang, B., Liu, S., Ma, A. J., Yip, T. C., et al. (2020). DATA-GRU: dual-attention time-aware gated recurrent unit for irregular multivariate time series. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, iaai 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020* (pp. 930–937). AAAI Press, URL https://ojs.aaai.org/index.php/AAAI/article/view/5440.

Tipirneni, S., & Reddy, C. K. (2022). Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data*, *16*(6), 105:1–105:17. http://dx.doi.org/10.1145/3516367.

Vapnik, V., Golowich, S., & Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. *Advances in Neural Information Processing Systems*, *9*.

Veillette, M. S., Samsi, S., & Mattioli, C. J. (2020). SEVIR : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural information processing systems 33: Annual conference on neural information processing systems 2020*. URL https://proceedings.neurips.cc/paper/2020/hash/fa78a16157fed00d7a80515818432169-Abstract.html.

Vodovotz, Y., An, G., & Androulakis, I. P. (2013). A systems engineering perspective on homeostasis and disease. *Frontiers in Bioengineering and Biotechnology*, *1*, 6.

Wang, Q., Chen, G., Jin, X., Ren, S., Wang, G., Cao, L., et al. (2023). Bit-MAC: Mortality prediction by bidirectional time and multi-feature attention coupled network on multivariate irregular time series. *Computers in Biology and Medicine*, *155*, Article 106586. http://dx.doi.org/10.1016/j.compbiomed.2023.106586.

Wang, Z., Liu, X., Huang, Y., Zhang, P., & Fu, Y. (2023). A multivariate time series graph neural network for district heat load forecasting. *Energy*, Article 127911.

Wang, Z., Zhang, Y., Jiang, A., Zhang, J., Li, Z., Gao, J., et al. (2021). Improving irregularly sampled time series learning with time-aware dual-attention memory-augmented networks. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 3523–3527).

Wanyan, T., Honarvar, H., Azad, A., Ding, Y., & Glicksberg, B. S. (2021). Deep learning with heterogeneous graph embeddings for mortality prediction from electronic health records. *Data Intelligence*, *3*(3), 329–339. http://dx.doi.org/10.1162/dint_a_00097.

Xu, F., & Tan, S. (2021). Deep learning with multiple scale attention and direction regularization for asset price prediction. *Expert Systems with Applications*, *186*, Article 115796. http://dx.doi.org/10.1016/j.eswa.2021.115796.

Yoon, J., Zame, W. R., & van der Schaar, M. (2019). Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, *66*, 1477–1490.

Zebin, T., Scully, P. J., & Ozanyan, K. B. (2016). Human activity recognition with inertial sensors using a deep learning approach. In *2016 IEEE sensors* (pp. 1–3). IEEE, https://ieeexplore.ieee.org/document/7808590.

Zhang, X., Zeman, M., Tsiligkaridis, T., & Zitnik, M. (2022). Graph-guided network for irregularly sampled multivariate time series. In *The tenth international conference on learning representations*. OpenReview.net, URL https://openreview.net/forum?id=Kwm8I7dU-I5.

Appendix 2

II

N. Bignoumba and S. Ben Yahiaand N. Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. In *Proceedings of KES'2024 - the 28th Annual KES Conference*, 2024

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

Deep Padding and Alignment Strategies for Irregular Multivariate Clinical Time Series

Nzamba Bignoumba[a,*], Sadok Ben Yahia[a,c], Nedra Mellouli[b,d]

[a] *Department of Software Science, Tallinn University of Technology, Akadeemia tee 15a, Tallinn 12618, Estonia*
[b] *ESILV, Léonrd de Vinci Group, Paris la Défense, France*
[c] *Centre for Industrial Software, University of Southern Denmark, Alsion 2, 6400- Sønderborg, Denmark*
[d] *LIASD, Université Paris 8, Paris, France*

Abstract

To improve the accuracy of an RNN when processing sparse and irregular multivariate clinical time series, we introduce two stacked deep learning models built on top of it, namely Padd-GRU and Alignment-driven Neural Network (ALNN). The Padd-GRU performs data-driven padding and imputation to obtain equal-length univariate and fill-in missing values, respectively. Then, the ALNN component transforms the resulting padded irregular multivariate clinical time series into a pseudo-aligned (or pseudo-regular) latent multivariate time series. We use the MIMIC-3 and PhysioNet databases to evaluate and compare our model to the state-of-the-art models on the mortality prediction task.

© 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the KES International.

Keywords: Alignment; Data-driven imputation; Classification; Irregular multivariate time series.

---

1. Introduction

Temporal irregularity is a common problem when processing data from one or more sensors with different temporal patterns of data collection [12]. This irregularity may also be due to data collection practices. When data are collected for a single feature/sensor, we face the problem of irregular univariate time series. On the other hand, when several features/sensors are involved, we speak of irregular multivariate time series. What makes temporal irregularity a problem in time series is that it leads to data sparsity, thereby increasing the rate of missing values. Recurrent neural networks (RNNs), which are among the best models for processing time series, have done well in many tasks [2, 13, 18], but they get stuck when they try to handle irregular time series. For example, the authors of [6] modified the RNN

---

[*] Corresponding author.
E-mail address: nzamba.bignoumba@taltech.ee

1877-0509 © 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the KES International.

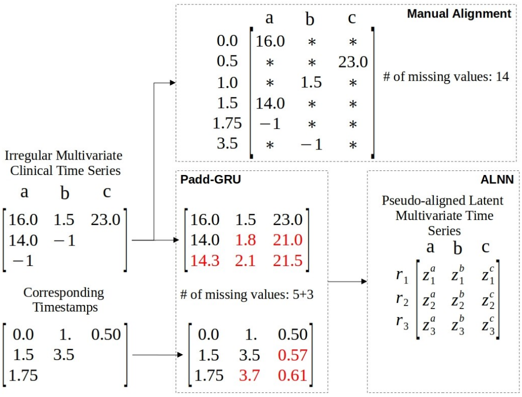


Fig. 1. Illustration of the Padd-GRU and ALNN processes. -1 represents missing values not due to temporal irregularity, while $*$ represents missing values caused by temporal irregularity. r_t are the even-spaced reference time points. Z_1^a is the latent value of the feature a at the reference time point r_1.

kernel to better accommodate irregular time series and improve the accuracy of mortality prediction, which is the task of interest in our study.

Several medical prediction tasks, especially the mortality prediction task, rely on multivariate clinical time series (MCTS). Due to variations in patient health status over time and the tools' data collection frequency, MCTS are often prone to spasticity (missing values) and temporal irregularities. Failing to implement specific strategies to overcome these pitfalls during decision-making, assisted by a prediction model, may lead to irreversible clinical events.

Unlike previous approaches that address the problem of irregular multivariate clinical time series (IMCTS) by: discretizing the data collection period [7, 24]; using ordinary differential equations (ODEs) to compute hidden states of RNNs in a continuous space [11], imputing/interpolating missing values caused by time irregularity within or outside the RNN core [5, 6, 15]; directly processing observed values and timestamps without additional imputation strategies [10, 21]; our method takes a different approach. It combines a data-driven padding technique with an alignment process, both applied on top of an RNN.

The goal of data-driven padding, which is done by Padd-GRU, a variant of the bidirectional recurrent imputation for time series (BRITS) [5], is to pad the shortest univariate time series so that all univariate time series have the same length. This padding strategy is also applied to the timestamp matrix. Additionally, Padd-GRU fills in missing values that are not due to temporal irregularity but to other problems, such as machine failure. The Padd-GRU output, which is the padded IMCTS, is then passed to the ALNN [3] for alignment. ALNN is a neural network designed to transform an irregular multivariate time series (IMTS) into a pseudo-aligned latent multivariate time series (PLMTS); see Figure 1 for illustration. The PLMTS is a matrix in which each row represents the latent values of each feature at each user-defined equally spaced reference time point. While ALNN has the advantage of not performing any manual alignment that is a noise generator (see manual alignment bloc in Figure 1 for illustration), it finds its limitation in the imputation strategies it applies to fill in missing values that are not due to temporal irregularity. In addition, these imputation strategies are based on strong assumptions. Therefore, we overcome this limitation with Padd-GRU, which performs data-driven imputation and fills in missing values based on the underlying structure of the data and the downstream task criterion. Overall, our model offers the following advantages over existing models:

- Performing a data-driven padding approach to obtain univariate time series of the same length and generate the corresponding timestamp of the padded values;
- Performing a data-driven imputation to fill in missing values that are not due to a temporal irregularity;
- Not carrying out manual alignment, which could generate noise.

Our motivation to conduct this study stems from the fact that many medical predictive studies rely on time series. In a sensitive field where human lives are at stake, it is critical to propose an effective approach to dealing with irregular clinical time series in order to avoid the worst. To summarize, the proposed model's main contributions are:

- The implementation of a data-driven imputation approach via Padd-GRU to fill in missing values;
- The implementation of a data-driven padding approach via Padd-GRU to handle univariate time series of different lengths;
- The transformation of IMCTS into PLMTS via the ALNN model;
- We use two publicly available databases, MIMIC-3 and PhysioNet, to compare the performance of our model to that of state-of-the-art models designed for irregular multivariate time series on the mortality prediction task. The result obtained shows that our model improves prediction accuracy.

2. Related works

In this section, we present relevant work related to irregular multivariate time series.

The main underlying problems encountered when performing classification, regression, or generation tasks based on multivariate time series data stem from how the data is collected. These include data sparsity, missing values, time series of different lengths, irregular timestamps, and misalignment. Concerning the irregular timestamps issue, which is one of the problems we are addressing in our study, the traditional approach to deal with it, is to discretize the period over which data are collected into evenly spaced time bins. Then, either static imputation methods such as mean and median imputations are used to fill in induced missing values during the pre-processing phase [7, 24], or a data-driven method is implemented in the model core to fill in missing values [5, 8, 19].

As this temporal discretization has the disadvantage of discarding fine-grained information and altering the temporal structure which can be a source of information, some methods directly deal with IMTS. Most of these methods are RNN-based models that have been redesigned to deal with IMTS. Indeed, the original RNN-based models are suboptimal when processing IMTS. For example, in [6], they proposed a GRU-based model that incorporates a decay mechanism to model temporal irregularity. Additionally, they performed a data-driven imputation. The authors in [20] also proposed a GRU-based model coupled with a dual-attention mechanism for the same purpose. In [11, 9], they proposed to compute the hidden state of RNN with ordinary differential equations (ODE) so that irregular time series can be processed in continuous latent space. Rather than modifying the structure of the RNN, the authors in [4] have built a Spline-based cubic neural network that interpolates irregular time series and projects them into a continuous latent space. Their representation in this continuous latent space is then divided into segments (to extract local features) and transmitted to an RNN. In [15], the authors developed a semi-parametric interpolation neural network that interpolates the IMTS against a set of evenly-spaced reference time points. This strategy allows them to transform IMTS into regular ones. Instead of using a set of evenly-spaced reference time points for interpolation purposes, which can be a source of noise, ALNN's authors [3] use them solely for alignment purposes. However, they still introduce noise into the model calculation due to the imputation performed during the pre-processing phase to fill in the initial missing values. Moreover, this imputation relies on strong assumptions.

Due to their high performance in natural language processing (NLP), Transformer architecture is increasingly adopted to handle IMTS [14, 17]. For example, the authors in [16], proposed an attention-based model composed of an encoder and decoder component. The encoder is responsible for encoding a set of latent variables conditioning a set of reference time points, the observed values and their corresponding irregular timestamps. On the other hand, the decoder is responsible for interpolating the missing values at a set of requested timestamps. In [21], they built a Transformer-based model that treats the time series as a set of tuples (observed value, feature, timestamp). These tuples are embedded in continuous vectors and pass through the Transformer component with multi-headed attention layers to learn contextual triplet embeddings. With this strategy, they do not need to perform any time discretization or alignment. In [23], they developed a time-aware dual-attention and memory-augmented networks model to overcome the problems of asynchronous interactions, time irregularity and data sparsity.

Since multivariate time series can be represented in graph form, several graph neural network-based models have also been proposed [1, 25, 22]. These models use a message-passing strategy to fill in the missing values from the observed values and the information associated with them.

Although the models mentioned above work well in practice, many of them require manual alignment (see Figure 1), which increases the rate of missing values and makes them suboptimal. To remedy this, in this study, we used ALNN, a deep neural network that was designed for alignment purposes while overcoming its limitations with Padd-GRU.

3. Method

When dealing with multivariate time series, we may be confronted with missing values caused by temporal irregularity. If the data already contained missing values at the outset, this temporal irregularity will increase the rate of missing values and therefore, make the downstream prediction/classification task more complex. We propose a data-driven padding approach, Padd-GRU, to mitigate the impact of missing values. Padd-GRU pads univariate time series of different lengths and fills in the initially missing values, those not due to temporal irregularity (unlike other types of missing values, their corresponding timestamp is recorded). The Padd-GRU's output, a padded IMTS, is subsequently fed into the ALNN to be transformed into a PLMTS. The ALNN transformation enables the application of state-of-the-art time series modelling algorithms, specifically RNNs, built for regular time series. In other words, Padd-GRU and ALNN will provide the RNN with a regular, imputed and padded version of the IMTS. Figure 2 gives an overview of the model's architecture. In what follows, we first introduce the data notation and then describe the different components of the proposed model.

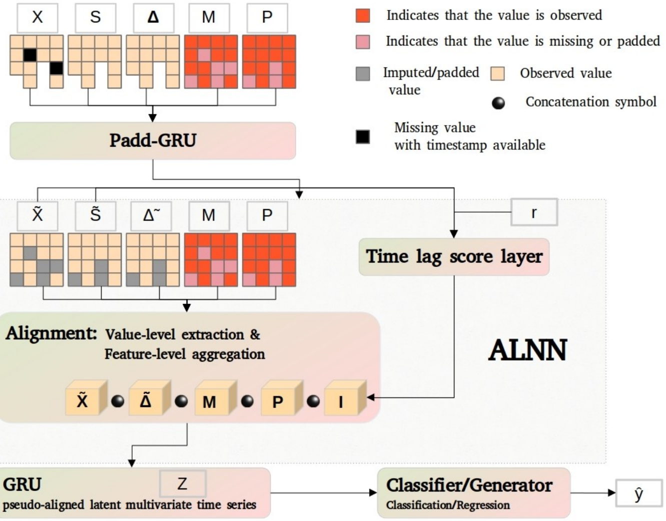

X
S
Δ
M
P
Indicates that the value is observed
Indicates that the value is missing or padded
Imputed/padded value
Observed value
Concatenation symbol
Missing value with timestamp available
Padd-GRU
X̃
S̃
Δ̃
M
P
r
Time lag score layer
Alignment: Value-level extraction & Feature-level aggregation
X̃
Δ̃
M
P
I
ALNN
GRU
Z
pseudo-aligned latent multivariate time series
Classifier/Generator
Classification/Regression
ŷ

Fig. 2. Model architecture

3.1. Data notation

Let $\mathcal{D} = \{X_n, M_n, P_n, S_n, \Delta_n, y_n\}_{n=1,2,\cdots,N}$ represents a dataset, where N is the number of samples. $X_n = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]_n^\top \in \mathbb{R}^{T\times K}$ is the *n-th* multivariate time series with T observations. K represents the number of features. $S_n = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_T]_n^\top \in \mathbb{R}^{T\times K}$, $M_n = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_T]_n^\top \in \{0,1\}^{T\times K}$, $P_n = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_T]_n^\top \in \{0,1\}^{T\times K}$ and $\Delta_n = [\delta_1, \delta_2, \cdots, \delta_T]_n^\top \in \mathbb{R}^{T\times K}$ are the timestamp matrix, the mask matrix, the padding matrix, and the time variation matrix associated with the *n-th* multivariate time series, respectively. The timestamp value for observation $x_t^k \in \mathbf{x}_t$ is s_t^k, and the time variation between two consecutive observations x_{t-1}^k and $x_{t>0}^k$ of the same univariate time series $\mathbf{x}_{:T}^k$ is $\iota_t^k \in \delta_t$, which is found in Equation 1. At $t = 0$, $\iota_t^k = 0$. $m_t^k \in \mathbf{m}_t$ is a binary value that indicates whether x_t^k is missing. $m_t^k = 1$ if x_t^k is observed; 0 otherwise. $p_t^k \in \mathbf{p}_t$ is a binary value that indicates whether x_t^k and s_t^k are padded values.

$p_t^k = 0$ if x_t^k and s_t^k are not observed; 1 otherwise. y_n is the target value of the *n-th* multivariate time series. Depending on the downstream task (classification/regression), y can be a categorical target or a real number.

$$\iota_t^k = \begin{cases} s_t^k - s_{t-1}^k + \iota_{t-1}^k & t > 1,\ m_{t-1}^k = 0 \\ s_t^k - s_{t-1}^k & t > 1,\ m_{t-1}^k = 1 \\ 0 & t = 1 \end{cases} \tag{1}$$

It is worth noting that since the univariate time series have different lengths at the outset, we first apply zero padding so that they can have the same T observations. We did the same for their corresponding timestamps and time variation matrices.

3.2. Data-driven padding: Padd-GRU

The data-driven padding module, Padd-GRU, is inspired by BRITS, a bidirectional recurrent neural network variant designed to fill in missing values in IMTS. BRITS combines a recurrent model with a regression model to impute missing values (if any) in each *t-th* observation. The bidirectional recurrent neural network performs this task in both forward and backward passes. BRITS can leverage the correlation between each variable to provide accurate imputed values (the reader may refer to [5] for more details). Formally, it is described as follows:

$$\hat{\mathbf{x}}_t = W_x \mathbf{h}_{t-1} + b_x \tag{2}$$

$$\hat{\mathbf{x}}_t^c = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{x}}_t \tag{3}$$

$$\gamma_t = exp\{-max(0, W_\gamma \delta_t + b_\gamma)\} \tag{4}$$

$$\hat{\mathbf{z}}_t = W_z \hat{\mathbf{x}}_t^c + b_z \tag{5}$$

$$\beta_t = \sigma(W_\beta[\gamma_t \circ \mathbf{m}_t] + b_\beta) \tag{6}$$

$$\hat{\mathbf{c}}_t = \beta_t \odot \hat{\mathbf{z}}_t + (1 - \beta_t) \odot \hat{\mathbf{x}}_t \tag{7}$$

$$\hat{\mathbf{c}}_t^c = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{c}}_t \tag{8}$$

$$\mathbf{h}_t = \sigma(W_h[\mathbf{h}_{t-1} \odot \gamma_t] + U_h[\hat{\mathbf{c}}_t^c \circ \mathbf{m}_t] + b_h) \tag{9}$$

$$l_t = \Big\langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) \Big\rangle + \Big\langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{z}}_t) \Big\rangle + \Big\langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{c}}_t) \Big\rangle \tag{10}$$

where $\odot$ and $\circ$ are the Hadamard product and the concatenation symbol, respectively. σ is a non-linear function. $\hat{\mathbf{x}}_t \in \mathbb{R}^K$, obtained by a linear transformation of the previous hidden state $\mathbf{h}_{t-1}$ is the estimation vector used to fill in missing values. At $t = 0$, $\mathbf{h}_{t-1} = \mathbf{0}$ where $\mathbf{0}$ is the zero vector. $\hat{\mathbf{x}}_t^c$ is the candidate observation vector. Its goal is to fill in missing values with those calculated in $\hat{\mathbf{x}}_t$ and keep unchanged the observed ones. γ_t is the decay factor and $\hat{\mathbf{z}}_t \in \mathbb{R}^K$ is a vector in which the value of each feature is calculated exclusively from those of the other ones. $\hat{\mathbf{z}}_t$ is called the "learnable correlation vector". $\hat{\mathbf{c}}_t$ is the weighted linear combination of $\hat{\mathbf{z}}_t$ and $\hat{\mathbf{x}}_t$. $\beta_t \in [0, 1]$ is the weighted factor. $\hat{\mathbf{c}}_t^c$ is the final candidate observation vector and $\mathbf{h}_t$ is the calculated hidden state of the *t-th* observation. W_* and b_* are the BRITS's learnable parameters. l_t is the loss accumulation function at each *t-th* observation. It uses $\mathbf{m}_t$ to calculate the loss exclusively from the observed values. $\mathcal{L}_e(.)$ is the mean absolute error.

In our case, we can use BRITS to fill in the missing initial values and obtain univariate values of the same length, but we need to redesign it to calculate the corresponding timestamps of the padded values. We calculated these timestamps because the ALNN needs them for alignment purposes. Unlike BRITS, we propose Padd-GRU, a unidirectional RNN that fills in the initial missing values, performs a padding to obtain a univariate variable of equal length, and then calculates the corresponding timestamps of the padded values.

Padd-GRU is a data-driven padding model. The padding it performs is based on the hidden structure of the data and the downstream task criterion. It introduces a learnable temporal variation vector in addition to the BRITS equations mentioned above and is calculated as follows:

$$\hat{\delta}_t = |W_\delta \mathbf{h}_{t-1} + b_\delta| \tag{11}$$

where $\hat{\delta}_t$ is the estimated time variation vector. We use the absolute value so that the values of the time variation vector can be restricted to positive values. W_δ and b_δ are learnable parameters. Since $\hat{\delta}_t$ is only used when processing padding values, the candidate time variation vector $\hat{\delta}_t^c$ is calculated as follows:

$$\hat{\delta}_t^c = \mathbf{p}_t \odot \delta_t + (1 - \mathbf{p}_t) \odot \hat{\delta}_t \tag{12}$$

where $\mathbf{p}_t$ is the current padding vector. Based on $\hat{\delta}_t^c$, we can then compute the new timestamp vector $\hat{\mathbf{s}}_t$, defined as follows:

$$\hat{\mathbf{s}}_t = \mathbf{p}_t \odot \mathbf{s}_t + (1 - \mathbf{p}_t) \odot (\mathbf{s}_{t-1} + \hat{\delta}_t^c) \tag{13}$$

where $\mathbf{s}_{t-1}$ is the timestamp vector of the (t-1)-*th* observation. Since the timestamp value cannot exceed the maximum duration of the data collection, we restrict the new values of the timestamp vector to values less than or equal to the maximum duration of the data collection. The restriction is formalized as follows:

$$\hat{s}_t^k = \begin{cases} s_t^k & p_t^k = 1 \\ s_{t-1}^k + (\hat{\iota}_t^k)^c & p_t^k = 0, [s_{t-1}^k + (\hat{\iota}_t^k)^c] \leq \Upsilon \\ s_{t-1}^k & p_t^k = 0, [s_{t-1}^k + (\hat{\iota}_t^k)^c] > \Upsilon \end{cases} \tag{14}$$

where $(\hat{\iota}_t^k)^c$ is a value of $\hat{\delta}_t^c$ and $\hat{s}_t^k$ is a value of $\hat{\mathbf{s}}_t$. Υ is the maximum duration of the data collection. For example, if the data was collected during 48 hours, the value of Υ will be 48. 48 is in fact the maximum value of timestamps. The next step is to update $\hat{\delta}_t^c$ as follows:

$$(\hat{\iota}_t^k)^c = \begin{cases} \iota_t^k & \mathrm{p}_t^k = 1 \\ \hat{\iota}_t^k & p_t^k = 0, [s_{t-1}^k + (\hat{\iota}_t^k)^c] \leq \Upsilon \\ \iota_{t-1}^k & p_t^k = 0, [s_{t-1}^k + (\hat{\iota}_t^k)^c] > \Upsilon \end{cases} \tag{15}$$

where ι_{t-1}^k is a value of δ_{t-1} (the time variation vector of the (t-1)-*th* observation). Since the decay factor (see Equation 4) depends on the time variation vector δ_t which has been redefined as $\hat{\delta}_t^c$, the new decay factor is now:

$$\gamma_t = exp\{-max(0, W_\gamma \hat{\delta}_t^c + b_\gamma)\} \tag{16}$$

To efficiently learn the time variation vector $\hat{\delta}_t$, we introduce the loss: $(\mathbf{p}_t, \mathcal{L}_e(\delta_t, \hat{\delta}_t)$. Therefore, the acumination loss at each *t-th* iteration becomes:

$$l'_t = \phi_1 l_t + \phi_2 \left\langle \mathbf{p}_t, \mathcal{L}_e(\delta_t, \hat{\delta}_t) \right\rangle \tag{17}$$

where l_t is the initial accumulation loss function presented in Equation 10. $\mathbf{p}_t$ is used to calculate only the loss generated by the padded values. The number of Padd-GRU units is set at 50 during the experiment. The regularization parameters ϕ_1 and ϕ_2 are both set at 0.5.

After the imputation and padding steps performed in Padd-GRU, the initial matrices of values X, timestamps S and time variations Δ are redefined as:

$$\tilde{X}_n = M_n \odot X_n + (1 - M_n) \odot \hat{C}_n^c; \tag{18}$$

$$\tilde{S}_n = P_n \odot S_n + (1 - P_n) \odot \hat{S}_n \tag{19}$$

$$\tilde{\Delta}_n = P_n \odot \Delta_n + (1 - P_n) \odot \hat{\Delta}_n^c \tag{20}$$

where $\hat{C}_n^c = [\hat{\mathbf{c}}_1^c, \hat{\mathbf{c}}_2^c, \cdots, \hat{\mathbf{c}}_T^c]_n^\top$, $\hat{S}_n = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \cdots, \hat{\mathbf{s}}_T]_n^\top$ and $\hat{\Delta}_n^c = [\hat{\delta}_1^c, \hat{\delta}_2^c, \cdots, \hat{\delta}_T^c]_n^\top$. $\tilde{X}, \tilde{S}$ and $\tilde{\Delta}$ will then be passed into the ALNN for alignment. In the following, we omit the subscript n for the simplicity of the formulas.

3.3. Alignment: ALNN

ALNN is a neural network designed to transform IMTS into PLMTS so that RNNs that are suboptimal with IMTS can be more accurate with PLMTS. The PLMTS is a matrix in which row values correspond to the latent value z_j^k of each feature k at regularly spaced user reference time points r_j where, $r_j \in \mathbf{r} = [r_1, r_2, \cdots, r_J]_{J=1,2,\dots}$. z_j^k is obtained through two phases including, value-level extraction and feature-level aggregation (due to page limit, readers may refer to [3] for more details).

Value-level extraction aims to non-linearly combine each $\tilde{x}_t^k \in \tilde{X}$ with its mask value m_t^k, time variation value $\tilde{\imath}_t^k \in \tilde{\Delta}$, and time lag score value $(i_t^k)^j$, to obtain more compact and richer information $(v_t^k)_j$. Additionally to $m_t^k, \tilde{\imath}_t^k$ and $(i_t^k)^j$, we also consider the corresponding padded indicator p_t^k. In contrast to the original ALNN, $(v_t^k)_j$ is then obtained as follows:

$$(v_t^k)_j = \sigma\{\tilde{x}_t^k(\breve{w}_1^k)_t^j + m_t^k(\breve{w}_2^k)_t^j + p_t^k(\breve{w}_3^k)_t^j + \tilde{\imath}_t^k(\breve{w}_4^k)_t^j + (i_t^k)_j(\breve{w}_5^k)_t^j + (\breve{b}^k)_t^j\} \tag{21}$$

$$(i_t^k)_j = \tilde{x}_t^k exp(-\gamma_j|r_j - \tilde{s}_t^k|) \tag{22}$$

where σ is a non-linear function. The time lag score $(i_t^k)^j$ represents the amount of information to consider from $\tilde{x}_t^k$ given the absolute time distance between its corresponding timestamps $\tilde{s}_t^k$ and the reference time point r_j. $(\breve{w}_*^k)_t^j$, $(\breve{b}^k)_t^j$ and γ_j are learnable parameters.

Feature-level aggregation aims to aggregate all $(v_t^k)_j$ values according to each feature k. Indeed, since we have J reference time points, then we will have J versions of each univariate time series as follows:

$$\mathbf{v}_1^k = [(v_1^k)_1, \cdots, (v_T^k)_1]^\mathsf{T}; \cdots; \mathbf{v}_J^k = [(v_1^k)_J, \cdots, (v_T^k)_J]^\mathsf{T} \tag{23}$$

where $\mathbf{v}_j^k$ is the latent time series of the feature k with respect to the reference time point r_j. For each aggregation performed in $\mathbf{v}_j^k$ we obtain:

$$z_j^k = \sigma(\sum_{i=1}^{T}(v_i^k)_l(\dot{w}_i^k)^j + (\dot{b}^k)^j);\ z_j^k \in Z \tag{24}$$

where z_j^k is the final latent value of the feature k at the reference time point r_j. $Z(z_j^k) \in \mathbb{R}^{J\times K}$ is the pseudo-aligned latent multivariate time series (PLMTS). It will be passed through a GRU for sequential modelling and subsequentially to the classifier.

3.4. GRU + Classifier

As GRU works better with regular time series than with irregular ones, ALNN provides it with PLMTS, which is a pseudo-regular version of the initial IMTS. The GRU output which is a context vector will then be fed to the classifier. The classifier encompasses two feedforward neural networks f_1 and f_2 with 100 and 1 units each, respectively. The activation function of f_1 is *relu* and for f_2 *sigmoid*. The model's output is then:

$$\hat{y} = f_{\theta_{f2}}^2 \circ f_{\theta_{f1}}^1 \circ GRU_{\theta_g}(Z) \tag{25}$$

where $\circ$ is the composition symbol. θ_g and θ_{f*} are the GRU and classifier parameters, respectively. $\hat{y} \in [0, 1]$ is the likelihood value. In addition to the losses introduced in Equation 17, we use the binary cross-entropy loss to calculate the classification error and update the model parameters. We use Adam as the optimizer with a learning rate set to 0.001. The number of GRU units is set at 168 during the experiment.

4. Experimental validation

Intensive care unit (ICU) mortality prediction can rely on irregular physiological time series data. That is why it is appropriate to use it as a pilot case to demonstrate the effectiveness of the proposed model. Furthermore, our main objective is to propose a decision-support model to practitioners to prevent adverse events in the ICU. Throughout this section, we conduct various experiments to evaluate the proposed model for ICU mortality prediction. We extract the datasets from two publicly available databases, MIMIC-3 and PhysioNet. The code is available at https://github.com/Zedfst/Padd_ALNN_GRU

4.1. Datasets

MIMIC-3 consists of anonymized health-related data from over forty thousand patients who stayed in the ICU at Beth Israel Deaconess Medical Center between 2001 and 2012. In this study, only patients who spend at least 48 are included. We obtained $27,162$ for patients fulfilling this condition. As some patients were admitted several times to the ICU, we extracted $32,496$ admissions distributed as follows: $28,075$ related to patients who remain alive and $4,421$ (15.75%) to patients who die. For each admission, 12 physiological time series data are used as model input. The associated target is a binary value, where 0 indicates that the actual physiological time series data is that of a still-alive patient and 1 that of a deceased patient.

PhysioNet is a database developed as part of a mortality prediction challenge. The database contains information on patients hospitalized for cardiac diseases in intensive care units. Only data from the first 48 hours after admission is available. We extracted data from $4,000$ patients, which is distributed as follows: $3,446$ related to patients who remain alive and 554 (13.85%) to patients who die. For each admission, 37 physiological time series data are used as model input. The associated target is a binary value, where 0 indicates that the actual physiological time series data is that of a still-alive patient and 1 that of a deceased patient.

4.2. Results

We devote this section to comparing and interpreting the performance of our model with that of competing models, including ALNN [3], Interp-net[15], GRU-D [6], mTAND [16], ODE-LSTM [9], Phased-LSTM [10] and STraTS [21]. All are described in Section 2. As we are dealing with unbalanced classes (alive and dead), we used the Area Under the ROC Curve (AUC) and the Area Under the Precision-Recall Curve (AUPRC) to evaluate the performance of our model against competing models over 5-fold cross-validation.

We conducted the experiment in two phases. The first consists of evaluating the models with the initial rate of missing values, which is 27.27% for MIMIC-3 and 27.91% for PhysioNet (see Table 1). In the second phase, we increase the missing value rate to 20% and 70% (see Table 2). We, therefore, obtain 47.27% and 97.27% for MIMIC-3 and 47.91% and 97.91% for PhysioNet.

Concerning the first experiment, which consists of evaluating the models with the initial rate of missing values, we see in Table 1 that the proposed model obtains the best performance. This highlights the effectiveness of the strategies we implemented. However, ALNN's AUC scores are similar, or almost similar, to ours. ALNN relies on the empirical mean to fill in some missing values, which explains this. Because there are more negative than positive classes, the empirical mean will tend to reflect the negative classes. Therefore, ALNN will use the empirical mean as a discriminative feature in multiple samples with missing values. The model's ability to correctly classify the negative class (patients alive) significantly influences the AUC score. We observe that the Phased-LSTM [10] model, which does not perform manual alignment but processes the IMTS directly as we do, is the third best-performing model. This confirms the need to avoid manual alignment, which can be a noise driver. We hypothesize that the ALNN and our model are better because of the data-driven alignment they perform. We suspect that Interp-net's [15] lower score is probably due to the manual alignment and interpolation it performs in its kernel (data-driven interpolation). This could introduce noise and alter the underlying structure of the data. We hypothesize that the data-driven imputation strategy used in GRU-D [6] works better and is less noisy than the one used in Phased-LSTM [10] because it relies on medical knowledge. However, manual alignment also has an impact on GRU-D's performance. Although mTAND [16] and STraTS [21] are transformer-based models, their scores are among the lowest. Indeed, with a relatively small amount of head attention, transformer-based models struggle to capture the complex hidden structure of IMTS.

In the second experiment, we increase the missing value rate to evaluate the robustness of our model. Table 2 shows that, even with a high number of missing values, compared with competing models, it remains the most accurate 7 out of 8 times.

5. Conclusion

We suggested adding two deep learning models on top of an RNN to make it more accurate when working with irregular multivariate time series data, especially clinical data. These models are Padd-GRU and ALNN. While the

Table 1. Models performance on the mortality prediction task (mean ± standard deviation from 5-fold cross validation).

	MIMIC-3		PhysioNet	
Model	AUC	AUPRC	AUC	AUPRC
ALNN [3]	0.848 ± 0.008	0.501 ± 0.024	**0.834 ± 0.013**	0.469 ± 0.029
Interp-net [15]	0.754 ± 0.005	0.313 ± 0.005	0.730 ± 0.020	0.330 ± 0.030
GRU-D [6]	0.826 ± 0.004	0.463 ± 0.019	0.806 ± 0.007	0.432 ± 0.024
mTAND [16]	0.807 ± 0.004	0.442 ± 0.017	0.720 ± 0.020	0.352 ± 0.017
ODE-LSTM [9]	0.806 ± 0.003	0.431 ± 0.016	0.766 ± 0.003	0.347 ± 0.031
Phased-LSTM [10]	0.831 ± 0.004	0.483 ± 0.021	0.818 ± 0.013	0.443 ± 0.033
STraTS [21]	0.800 ± 0.010	0.429 ± 0.003	0.730 ± 0.016	0.354 ± 0.011
Ours	**0.850 ± 0.010**	**0.510 ± 0.020**	**0.834 ± 0.007**	**0.478 ± 0.018**

Table 2. Model performance when the rate of the missing value is increased.

		MIMIC-3		PhysioNet	
+ missing value rate	Model	AUC	AUPRC	AUC	AUPRC
+20%	ALNN [3]	0.839 ± 0.010	**0.480 ± 0.025**	0.827 ± 0.014	0.459 ± 0.035
	Interp-net [15]	0.640 ± 0.002	0.252 ± 0.017	0.613 ± 0.012	0.245 ± 0.010
	GRU-D [6]	0.775 ± 0.014	0.362 ± 0.016	0.788 ± 0.013	0.392 ± 0.022
	mTAND [16]	0.768 ± 0.012	0.360 ± 0.002	0.710 ± 0.020	0.350 ± 0.020
	ODE-LSTM [9]	0.695 ± 0.011	0.339 ± 0.007	0.618 ± 0.038	0.244 ± 0.014
	Phased-LSTM [10]	0.791 ± 0.020	0.426 ± 0.027	0.807 ± 0.017	0.438 ± 0.035
	STraTS [21]	0.719 ± 0.016	0.346 ± 0.008	0.728 ± 0.023	0.349 ± 0.015
	Ours	**0.840 ± 0.010**	**0.480 ± 0.020**	**0.830 ± 0.008**	**0.472 ± 0.018**
+70%	ALNN [3]	**0.805 ± 0.010**	0.390 ± 0.022	**0.803 ± 0.014**	0.413 ± 0.037
	Interp-net [15]	0.621 ± 0.002	0.244 ± −0.001	0.599 ± 0.017	0.240 ± 0.012
	GRU-D [6]	0.738 ± 0.017	0.312 ± 0.025	0.781 ± 0.014	0.387 ± 0.029
	mTAND [16]	0.760 ± 0.022	0.355 ± 0.018	0.700 ± 0.020	0.350 ± 0.010
	ODE-LSTM [9]	0.673 ± 0.013	0.328 ± 0.011	0.599 ± 0.044	0.239 ± 0.018
	Phased-LSTM [10]	0.752 ± 0.022	0.371 ± 0.037	0.735 ± 0.028	0.316 ± 0.065
	STraTS [21]	0.720 ± 0.017	0.346 ± 0.008	0.735 ± 0.013	0.352 ± 0.010
	Ours	0.803 ± 0.011	**0.399 ± 0.026**	**0.803 ± 0.011**	**0.418 ± 0.017**

Padd-GRU is responsible for filling in the missing values and padding the univariate time series so that they have the same number of observations, ALNN transforms the output of the Padd-GRU into a pseudo-aligned latent multivariate time series. Therefore, RNN can handle the pseudo-aligned latent multivariate time series more effectively than the initial irregular multivariate time series. As mortality prediction can be based on irregular multivariate clinical time series, we have chosen it as a pilot case to compare and evaluate our model with the state-of-the-art models designed to deal with irregular multivariate time series. The AUC and AUPRC scores indicate that our model handles irregular multivariate clinical time series better. We are optimistic about its ability to help practitioners make decisions in the ICU to reduce mortality.

One of the main limitations of your model is its high complexity (space and time). For instance, training takes +7 minutes compared with the original ALNN. However, the test duration is almost the same. We believe this can be mitigated by reducing the reference time point and making them learnable parameters to preserve model performance. Although we used two different databases, the model was evaluated on the same pilot case. In addition, we did not provide an explicable component, which is crucial in sensitive domains such as medicine. Therefore, future work should focus on reducing complexity, providing model explainability, and evaluating the model on broader clinical and non-clinical tasks. Its use in non-clinical tasks will enable us to assess its generalizability.

References

[1] Andrea, C., Ivan, M., Alippi, C., et al., 2021. Filling the g_ap_s: Multivariate time series imputation by graph neural networks, in: ICLR 2022, pp. 1–20.

[2] Bertl, M., Bignoumba, N., Ross, P., Yahia, S.B., Draheim, D., 2024. Evaluation of deep learning-based depression detection using medical claims data. Artificial Intelligence in Medicine 147, 102745.

[3] Bignoumba, N., Mellouli, N., Yahia, S.B., 2024. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. Expert Systems with Applications 238, 122148.

[4] Bilos, M., Ramneantu, E., Günnemann, S., 2022. Irregularly-sampled time series modeling with spline networks. CoRR abs/2210.10630. URL: `https://doi.org/10.48550/arXiv.2210.10630`, doi:`10.48550/ARXIV.2210.10630`, `arXiv:2210.10630`.

[5] Cao, W., Wang, D., Li, J., Zhou, H., Li, L., Li, Y., 2018. Brits: Bidirectional recurrent imputation for time series. Advances in neural information processing systems 31.

[6] Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. Scientific reports 8, 1–12.

[7] El-Rashidy, N., El-Sappagh, S., Abuhmed, T., Abdelrazek, S., El-Bakry, H.M., 2020. Intensive care unit mortality prediction: An improved patient-specific stacking ensemble model. IEEE Access 8, 133541–133564.

[8] Jun, E., Mulyadi, A.W., Choi, J., Suk, H.I., 2020. Uncertainty-gated stochastic sequential model for ehr mortality prediction. IEEE Transactions on Neural Networks and Learning Systems 32, 4052–4062.

[9] Lechner, M., Hasani, R.M., 2020. Learning long-term dependencies in irregularly-sampled time series. CoRR abs/2006.04418. URL: `https://arxiv.org/abs/2006.04418`, `arXiv:2006.04418`.

[10] Neil, D., Pfeiffer, M., Liu, S.C., 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. Advances in neural information processing systems 29.

[11] Rubanova, Y., Chen, R.T.Q., Duvenaud, D., 2019a. Latent odes for irregularly-sampled time series. CoRR abs/1907.03907. URL: `http://arxiv.org/abs/1907.03907`, `arXiv:1907.03907`.

[12] Rubanova, Y., Chen, T.Q., Duvenaud, D., 2019b. Latent ordinary differential equations for irregularly-sampled time series, in: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 5321–5331. URL: `https://proceedings.neurips.cc/paper/2019/hash/42a6845a557bef704ad8ac9cb4461d43-Abstract.html`.

[13] Sagheer, A., Kotb, M., 2019. Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323, 203–213.

[14] Shan, S., Li, Y., Oliva, J.B., 2023. NRTSI: non-recurrent time series imputation, in: IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023, IEEE. pp. 1–5. URL: `https://doi.org/10.1109/ICASSP49357.2023.10095054`, doi:`10.1109/ICASSP49357.2023.10095054`.

[15] Shukla, S.N., Marlin, B.M., 2019. 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. URL: `https://openreview.net/forum?id=r1efr3C9Ym`.

[16] Shukla, S.N., Marlin, B.M., 2021. Multi-time attention networks for irregularly sampled time series, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net. URL: `https://openreview.net/forum?id=4c0J6lwQ4_`.

[17] Song, H., Rajan, D., Thiagarajan, J., Spanias, A., 2018. Attend and diagnose: Clinical time series analysis using attention models, in: Proceedings of the AAAI conference on artificial intelligence.

[18] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D., 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, Anchorage AK USA. pp. 2828–2837. URL: `https://dl.acm.org/doi/10.1145/3292500.3330672`, doi:`10.1145/3292500.3330672`.

[19] Suo, Q., Zhong, W., Xun, G., Sun, J., Chen, C., Zhang, A., 2020. Glima: Global and local time series imputation with multi-directional attention learning, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE. pp. 798–807.

[20] Tan, Q., Ye, M., Yang, B., Liu, S., Ma, A.J., Yip, T.C.F., Wong, G.L.H., Yuen, P., 2020. Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series, in: Proceedings of the AAAI conference on artificial intelligence, pp. 930–937.

[21] Tipirneni, S., Reddy, C.K., 2022. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. ACM Transactions on Knowledge Discovery from Data (TKDD) 16, 1–17.

[22] Wang, D., Yan, Y., Qiu, R., Zhu, Y., Guan, K., Margenot, A., Tong, H., 2023. Networked time series imputation via position-aware graph enhanced variational autoencoders, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2256–2268.

[23] Wang, Z., Zhang, Y., Jiang, A., Zhang, J., Li, Z., Gao, J., Li, K., Lu, C., Ren, Z., 2021. Improving irregularly sampled time series learning with time-aware dual-attention memory-augmented networks, in: Proceedings of the 30th ACM international conference on information & knowledge management, pp. 3523–3527.

[24] Wanyan, T., Honarvar, H., Azad, A., Ding, Y., Glicksberg, B.S., 2021. Deep learning with heterogeneous graph embeddings for mortality prediction from electronic health records. Data Intelligence 3, 329–339.

[25] Zhang, X., Zeman, M., Tsiligkaridis, T., Zitnik, M., 2021. Graph-guided network for irregularly sampled multivariate time series. arXiv preprint arXiv:2110.05357 .

Appendix 3

III

M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, and D. Draheim. Evaluation of deep learning-based depression detection using medical claims data. *Artificial Intelligence in Medicine*, 147:102745, 2024

Artificial Intelligence In Medicine 147 (2024) 102745

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research paper

Evaluation of deep learning-based depression detection using medical claims data

Markus Bertl [a,*], Nzamba Bignoumba [b], Peeter Ross [a,c], Sadok Ben Yahia [b,d], Dirk Draheim [e]

[a] *Department of Health Technologies, Tallinn University of Technology, Akadeemia Tee 15A, Tallinn, 12618, Estonia*
[b] *Department of Software Science, Tallinn University of Technology, Akadeemia Tee 15A, Tallinn, 12618, Estonia*
[c] *Department of Research, East-Tallinn Central Hospital, Ravi 18, Tallinn, 10138, Estonia*
[d] *University of Southern Denmark, Alsion 2, Sønderborg, 6400, Denmark*
[e] *Information Systems Group, Tallinn University of Technology, Akadeemia Tee 15A, Tallinn, 12618, Estonia*

ARTICLE INFO

Keywords:
Artificial intelligence (AI)
Decision support system (DSS)
Deep learning
Machine learning (ML)
Depression
Insurance data
Medical claims data
Psychiatry

ABSTRACT

Human accuracy in diagnosing psychiatric disorders is still low. Even though digitizing health care leads to more and more data, the successful adoption of AI-based digital decision support (DDSS) is rare. One reason is that AI algorithms are often not evaluated based on large, real-world data. This research shows the potential of using deep learning on the medical claims data of 812,853 people between 2018 and 2022, with 26, 973, 943 ICD-10-coded diseases, to predict depression (F32 and F33 ICD-10 codes). The dataset used represents almost the entire adult population of Estonia. Based on these data, to show the critical importance of the underlying temporal properties of the data for the detection of depression, we evaluate the performance of non-sequential models (LR, FNN), sequential models (LSTM, CNN-LSTM) and the sequential model with a decay factor (GRU-Δt, GRU-decay). Furthermore, since explainability is necessary for the medical domain, we combine a self-attention model with the GRU decay and evaluate its performance. We named this combination Att-GRU-decay. After extensive empirical experimentation, our model (Att-GRU-decay), with an AUC score of 0.990, an AUPRC score of 0.974, a specificity of 0.999 and a sensitivity of 0.944, proved to be the most accurate. The results of our novel Att-GRU-decay model outperform the current state of the art, demonstrating the potential usefulness of deep learning algorithms for DDSS development. We further expand this by describing a possible application scenario of the proposed algorithm for depression screening in a general practitioner (GP) setting—not only to decrease healthcare costs, but also to improve the quality of care and ultimately decrease people's suffering.

1. Introduction

Psychiatric disorders, especially mood disorders such as depression, represent the critical non-communicable diseases of the 21st century and are ranked as the leading cause of years lived with disabilities [1]. Unfortunately, these diseases are often diagnosed late or incorrectly [2,3]. According to [4], depression is diagnosed by general practitioners with a sensitivity of 50.1% (95% CI: 41.3 to 59.0) and a specificity of 81.3% (95% CI: 74.5 to 87.3). Other research suggests that in the US, two-thirds of depression patients go undiagnosed [5]. Psychiatric diseases affect people's health and leave an impact from a cost perspective on a more global level. In Europe alone, psychiatric disorders accounted for EUR 461 billion in healthcare costs [6]. Of all psychiatric diseases, the economic costs of depression are among the highest. Other research suggests that the quality-adjusted life years (QUALYs) lost amount to $9950 per citizen with undiagnosed depression [7].

The recommended method for diagnosing depression in 2023 is based on questionnaires and assessment scales from the previous century [8]. In psychology and psychiatry, medical professionals still rely on methods dating back to the 1960s [9,10]. New technologies such as Artificial Intelligence (AI), especially deep learning, could potentially improve this situation by supporting medical professionals. The computer-based systems that use data to assist decision-making are called *digital decision support systems* (DDSS). AI-based DDSS for psychiatry is an active research field [11,12]. However, research often does not make its way into clinical practice [13]. One cause is the missing domain understanding among the DDSS developers and the heterogeneity of domain understanding among the DDSS developers and domain experts [14]. Another reason is that the data used to develop such systems are often unavailable or of bad quality [15]. More and more countries are applying a single public payer approach to health care,

---

[*] Corresponding author.
E-mail address: mbertl@taltech.ee (M. Bertl).

https://doi.org/10.1016/j.artmed.2023.102745
Received 14 May 2023; Received in revised form 1 December 2023; Accepted 1 December 2023
Available online 4 December 2023
0933-3657/© 2023 Elsevier B.V. All rights reserved.

like Canada [16], Australia [17], the UK [18] or Estonia [17], meaning that a vast amount of medical claims data will be available in a central place. However, data are mostly used for claims management and rarely reused. This paper investigates which algorithm is best suited for building a deep learning-based DDSS for depression detection based on medical claims data. As one of the leaders of e-government [19], Estonia is a good starting point for such research because lots of digital data are already available.

The foundation of the Estonian e-state is its digital identity system [20]. Each of the 1.3 million residents of Estonia has their own unique ID code. This allows for the creation of digital government services like online income tax declaration (used by 96% of people [21]), internet voting (used by 46.7% [22]) or e-prescription (used by 99% [23]) — e-health services especially profit from the Estonian e-state. One example is the collection of medical data. In Estonia, medical data are saved in two central places. The first is the e-health system called the Estonian Nationwide Health Information System (NHIS). The NHIS has been operational since 2008, allowing secure and trusted online access to medical data, prescriptions and medical images for virtually all Estonian residents. Instead of one big centralized database, the NHIS comprises several federated and mutually independent systems. One of them is the nationwide electronic health record (EHR) system. In the central EHR, patient data are saved based on international standards such as HL7 CDA,[1] DICOM,[2] LOINC,[3] ICD-10[4] and SNOMED-CT.[5]

The second place is the Estonian Health Insurance Fund (EHIF), which manages healthcare expense reimbursement. Their digital system was introduced in 2001 as an addition to the paper-based process. Since 2005, all reimbursement claims and prescriptions must be submitted electronically. As of today, the data collection process of the EHIF is part of their reimbursement process for healthcare providers. Medical professionals fill in the case history and demographic data in a structured electronic medical record system. Then they compose a discharge letter or an outpatient summary, where the diseases are coded using ICD-10 codes. This information is then sent to the EHIF, where it is automatically quality-checked and a random sample set of cases is manually validated before the reimbursement process is initiated. After the checks, the data are saved in the EHIF data warehouse. The medical statisticians of the EHIF then use the data for research, political or healthcare policy decision-making or to supply information to other governmental authorities. Based on the role of the data requester, the data are available in either personalized form with patient identifiers or in anonymized form. In the case of Estonia, a duplication of the data is also saved to the NHIS.

Since ICD codes are a concatenation of digits and alphabetic characters that convey information, they are considered categorical features. Thus, for analysis and prediction purposes involving ICD codes, we can benefit from the state-of-the-art machine and deep learning models dedicated to NLP tasks, such as [24–26]. For our work, we leverage on a *self-attention layer*,[6] which is a *transformer's sub-layer* [25], for efficient encoding of hidden relationships between diagnoses.

Since a single data modality (in our case, diagnosis) is usually not consistent enough for effective decision-making, it is common in the medical field to merge heterogeneous features or homogeneous features with different modalities, such as clinical text, demographics, images, or IoT sensor data [27–30]. In the case of depression detection, due to its heterogeneity, i.e. different types of depression, several heterogeneous data are usually involved. Indeed, the high number of similarities that exist between some depression types make their classification complicated. Although considering several heterogeneous data may improve model accuracy, some are difficult to collect or biased with inconsistent patient responses. Assuming that diagnoses performed by a doctor are more trusted and accessible, we decided to combine them with demographic data for more accurate depression detection.

Medical events are recorded with their corresponding date in the patient's electronic health record (EHR). The recording date plays a crucial role in the clinical process: it allows practitioners to track the trajectory of the patient's health status over time to make appropriate decisions. The omission of this information for decision support will undoubtedly result in lower performance and make the model less realistic. It is therefore necessary to consider the sequence of medical events. Like many models that have been built for health care (or used a medical problem as a pilot case) [31,32], we also consider the time intervals between consecutive diagnoses as an additional input for our model so that the process of detecting depression can rely more on recent diagnoses. As in [31,33], we model this temporal aspect by incorporating a decay factor in the gated recurrent unit (GRU) [34]. Considering the time intervals between consecutive diagnoses as additional inputs and effectively incorporating them into the GRU's core via a decay factor sets our proposal apart from previous works on depression detection [35].

Depression is often addressed like a binary classification or multimodal logistic regression problem [36,37]. Binary classification determines whether a patient suffers from depression, while multimodal logistic regression associates each type of depression with a probability score. The highest scoring type is then selected as the diagnosis. Although multimodal logistic regression has the advantage of learning the distribution of each depression type mutually, it faces the problem of imbalanced class distribution. Moreover, only patients who suffer from depression are studied. As we want to minimize imbalanced class problems, detecting patients suffering from depression and those not suffering, we choose the binary classification approach with class-weighting factors. Unlike our predecessors, here are additional aspects that we considered:

- Using a *self-attention layer* to effectively learn hidden relationships between diagnoses to better represent patient health status.
- Weighting the significance of diagnoses based on their corresponding record date so that the model can rely more on recently made diagnoses.
- Using *weighted binary cross-entropy* as the loss function to deal with the imbalanced class problem.
- Comparing the performance of non-sequential models, sequential models and sequential models with a decay factor versus our novel approach to show the importance of good encoding of the hidden relationship between diagnoses and the importance of considering the time factor.
- Integrating an explainable component so that physicians have greater confidence in the decision made by the model.

With the proposed approach, we aim to provide an AI model that helps medical professionals to overcome the challenge of low diagnostic accuracy. To improve the diagnostic process, we not only propose a deep learning approach with sufficient accuracy, but we also ensure that our algorithm is trained on a sufficiently large quantity of real-world data that is available during the clinical process and propose an application scenario for our algorithm.

The remainder of this paper is organized as follows: In Section 2, we present background works. In Section 3, we formally represent the dataset and describe our model. Section 4 is devoted to empirically evaluating our model against our competitors on different metrics. We have also carried out various ablation studies to show how the

---

[1] http://www.hl7.org/.

[2] https://www.dicomstandard.org/.

[3] https://loinc.org/.

[4] https://icd.who.int/browse10/.

[5] https://www.snomed.org/.

[6] The word layer can be used interchangeably with block, model or component.

model works in different configurations. In Section 6, we discuss the explainability of the results. In Section 7, we discuss possible use case scenarios, limitations and further research. Section 8 recalls the paper's main points and contributions and outlines future work.

2. Related work

Data science aims to develop computational models that can automatically infer hidden patterns from data to predict results. Predictions can be based on single or multi-modal data sources [38]. For depression detection, several data sources like audio [39–41], EEGs [42–45], IoT or wearable data [46–49], medical images [50,51] and text data [52–55] have been investigated. However, these data must be specifically collected and available for a decision support system. We argue that data generated during the clinical process (like diagnosis data) have a much higher chance to power successful DDSSs because data availability and quality are lower and privacy issues are fewer. Examples of these data include medical claims or electronic health record data. Promising results on using medical claims data for calculating the risk of suicide prevention have been reported in [28]. Medical claims data are also used to predict reactions to antidepressant treatment [56]. However, studies using machine learning [36] or rule-based approaches [57] on medical claims data for depression screening still report low accuracy metrics.

With a growing volume of data and features and increased computing power, deep learning starts to outperform traditional ML methods [58]. Traditional ML methods typically require good feature selection and a significant amount of feature engineering to ensure that the features used comply with the model's assumptions. On the other hand, deep learning uses a large, multi-layer network structure, allowing it to take raw input features and still be able to learn hidden patterns in data. Deep learning architectures can be distinguished by the structure determining how the network's artificial neurons are connected. For processing sequential and/or structured/unstructured data (like historical diagnoses and medication, clinical notes and images), *recurrent neural networks* (RNNs) [59], *convolutional neural networks* (CNNs) [60], *transformers* [25] and *graph neural networks* (GNNs) [61] are perfect candidates. Due to their high performance on non-medical tasks addressed with the same data structure as medical applications, their utilization in the medical field has increased significantly. For example, in [62], GNNs are combined with a pre-trained transformer-based model, namely BERT (*Bidirectional Encoder Representations from Transformers*) [63] for medical code representation and medication recommendation. Also, in [64], a pre-trained BERT (specifically its transformer-encoder component) is used to predict 30-day hospital readmissions from clinical notes. In [65], a cost-sensitive formulation of *long short-term memory networks* (LSTM) [66] is proposed to predict 30-day readmission of congestive heart failure patients. Similar work using machine learning and deep learning approaches to predict mortality and readmission of in-hospital cardiac arrest patients with EHR was also conducted in [67]. In [68], a convolutional graph transformer is developed to learn the hidden structure of Electronic Health Record (EHR) data for graph reconstruction while predicting hospital readmission. While some minor modifications had to be made to the core of the aforementioned deep learning models to deal with medical data efficiently, even more significant changes were needed to deal with the ubiquitous irregular time series in the medical field. For example, several studies [31,33,69] have focused on redesigning RNNs to better handle irregular physiological time series data and thus improve the accuracy of downstream medical tasks.

Regarding our main concern, namely the detection of depression, we note several studies based on deep learning methods, such as [70–72], which have used social network data rather than medical claims data. For example, in [73], an LSTM-based model is coupled with an attention mechanism to detect depression from users' tweets. The dataset was balanced (oversampling or undersampling) to address the imbalanced class problem. In [74], a data augmentation framework based on topic modelling is proposed to solve the problem of imbalanced classes when detecting depression. In contrast to approaches based on social network data, [74] uses patients' responses recorded during encounters with doctors. Unlike the technique we use (the cost-sensitive loss function), the undersampling or oversampling technique has the disadvantage of altering the natural distribution of the data used in the study. In addition, they may lose some information or add noise. Although several depression detection studies have been conducted using social network data, some, like ours, have used medical claims data [75,76]. In [35], a bidirectional deep learning model is proposed—a pre-trained and fine-tuned version of the BERT model. Compared to our proposal, where only diagnoses and patient demographics are used, the approach in [35] uses additional modalities such as procedures, medications and clinical notes. Our approach, as in [35], relies on the self-attention component to quantitatively assess the association between clinical codes; however, it does not consider the time interval when modelling consecutive visits.

As we can see, none of the aforementioned models addressing the problem of depression detection have combined *self-attention* with GRU-decay for better data representation and efficient integration of temporal information, respectively. Additionally, unlike some, we use real and voluminous medical datasets and do not apply any undersampling or oversampling that may remove relevant information or introduce noise. Instead, we use a cost-sensitive loss function to deal with the problem of imbalanced classes.

3. Method

Detecting depression from claims data involves considering three important aspects: learning the hidden relationships between diagnoses; filtering out the irrelevant diagnoses; and relying more on recent diagnoses. In addition, we may face an imbalanced class problem, as there are naturally fewer sick patients than healthy ones.

In this section, after introducing data notation, we formally describe how a self-attention layer is stacked with GRU-decay to cover the aforementioned aspects. Self-Attention is dedicated to learning hidden relationships across diagnoses and filtering out the irrelevant ones. At the same time, GRU-Decay allows detection based on the most recent diagnoses. We name this combination Att-GRU-decay. The output of Att-GRU-decay, which is the global health status of the patient, is combined with the patient's demographics and fed into a classifier that we also describe formally in the following subsections. Finally, we present the loss function used to address the imbalanced class problem. The overall model architecture is depicted in Fig. 1.

3.1. Data notation

Let $\mathcal{D} = \{\mathbf{c}_n, \mathbf{d}_n, \Delta_n, y_n\}_{n=1,2,\ldots,N}$ where $\mathbf{c}_n = [c_1, c_2, \ldots, c_{t=T}]$ is the set of diagnoses (ICD-10 codes) of the patient n recorded at date index $t = 1, 2, \ldots, T$. If a patient suffers from depression, the highest date index T is the one preceding the date on which the depression was detected in the patient. Otherwise, the highest date index T is that of the last diagnosis made. $\mathbf{d}_n = [d_1, d_2]$ is the patient's demographic vector. d_1 is the age and d_2 is the gender. $\Delta_n = [\delta_1, \delta_2, \ldots, \delta_{t=T}]$ is the elapsed time vector. More precisely, δ_t with $t > 1$ and $t < T$ is the time difference between the recorded date of the medical code c_t and c_{t-1}. $\delta_1 = \delta_T = 1$. y_n is the depression state of the patient n: equal to 0 if the patient has never suffered from depression; otherwise, it is equal to 1.

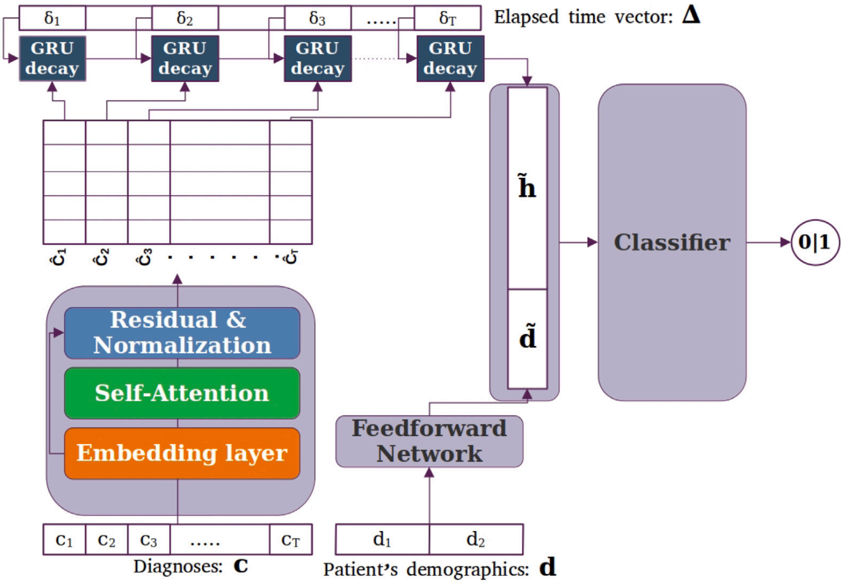


Fig. 1. Att-GRU-decay architecture. The embedding layer encodes diagnoses into continuous vectors; Self-attention learns the hidden relationship between the diagnoses pair; and the Residual & Normalization retain the initial information and prevent gradient problems. GRU-decay learns the sequential pattern of diagnoses while taking into account the elapsed time between visits δ_t and generates a context vector $\tilde{h}$, which is a latent representation of the patient's health status. $\tilde{h}$ is concatenated with the latent representation of the patient's demographics and passed through the classifier.

3.2. Self-attention

This section presents how diagnoses are transformed into a vector embedding and passed through a self-attention layer responsible for learning hidden relationships between diagnoses and filtering out the ones irrelevant to the downstream task.

Since neural networks require real numbers as inputs, the first step is to map each diagnosis to a vector of real numbers. For that, we use an *embedding layer*, which, based on the co-occurrence of diagnoses, will associate with each diagnosis c_t[7] a vector embedding $\tilde{\mathbf{c}}_t$ obtained as part of a matrix $\tilde{C}$ of all vector embeddings as follows:

$$\tilde{C} = Embedding_{\theta}(\mathbf{c}) \tag{1}$$

where θ are the learnable parameters of the *embedding layer*. $\tilde{C} \in \mathbb{R}^{T\times l}$ is the matrix of diagnoses encoded, where each row t of $\tilde{C}$ is the vector embedding $\tilde{\mathbf{c}}_t$ of the diagnosis c_t, and l is the dimension of the embedding space.

To discover the latent relationships between diagnoses and filter out diagnoses that might not be relevant for detecting whether a patient suffers from depression, we pass $\tilde{C}$ through a self-attention layer. Throughout self-attention computations, we calculate an attention filter from a query matrix Q and a key matrix K to encode hidden relationships between diagnoses. This attention filter is then multiplied by a value matrix V to obtain a filtered version $\tilde{C}'$ of $\tilde{C}$. In other words, those vector embeddings $\tilde{\mathbf{c}}_t$ that will be detected as irrelevant for determining whether a patient suffers from depression will have coefficient values close to zero. The formula for calculating $\tilde{C}'$ is

$$\tilde{C}' = \underbrace{Softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)}_{attention filter} V \tag{2}$$

where $Q = \tilde{C}W_Q \in \mathbb{R}^{T\times j}$, $K = \tilde{C}W_K \in \mathbb{R}^{T\times j}$ and $V = \tilde{C}W_V \in \mathbb{R}^{T\times j}$ are three different linear transformations of $\tilde{C}$. W_Q, W_K and W_V are learnable parameters, $j = l$ is the dimension of each linear space, d_k is the dimension of the key vectors, and as usual, $K^{\top}$ stands for the transpose of K. $\tilde{C}'$ is normalized to prevent exploding values. The normalized version of $\tilde{C}'$ is then added to $\tilde{C}$ to preserve initial relevant information that might be lost during self-attention computation and to prevent gradient problems. Residual is the sum of an input x with the output $y = f(x)$ [77]. The final output of the self-attention layer is then equal to:

$$\hat{C} = \underbrace{\tilde{C} + Normalize(\tilde{C}')}_{residual} \tag{3}$$

$\hat{C} \in \mathbb{R}^{T\times j}$ is a matrix, where each row t is the final embedding representation $\hat{\mathbf{c}}_t$ of a corresponding diagnosis c_t.

As some diagnoses may have been made long ago, assessing their significance in terms of when they were made is crucial. Thus, in Section 3.3, we formally show how we apply a decay factor on the hidden layer of GRU so that past diagnoses cannot have the same level of importance as recent ones.

It is worth mentioning that self-attention aims to encode the hidden correlation between pairs of clinical codes, while GRU-decay encodes the sequential order of visits, taking into account the time elapsed between them. We could have used the positional encoding technique implemented in the original Transformer [25] to model the sequential order of visits. However, since the positional encoding vectors are static, we would not have been able to capture the variation in elapsed time between successive visits effectively. Shaw et al. [78], proposed a Transformer variant capable of modelling the relative position or distance between input element pairs. However, the method they use to achieve this lacks the property of the decay function implemented in our model (see Eq. (8)). Indeed, the objective is not only to encode the distance (in terms of days) between input element pairs. It is also about reducing the impact of the latter in the prediction when they took place a long time ago.

---

[7] As the following formulas are valid for all patients, we omit the subscript n in the sequel.

3.3. GRU-decay

Although some patients may suffer from depression without prior symptoms, we can detect those who do with specific earlier symptoms. This then requires browsing the patient's historical diagnoses. Since diagnoses are described by a set of clinical codes recorded over time, depression detection can be approached as both time-series forecasting and NLP tasks.

With RNNs and their variants having shown spectacular results on time series forecasting and NLP tasks, e.g. [79,80], we can use them to model the patient's status while considering the time at which each diagnosis has been recorded. Since RNNs suffer from gradient problems when processing long sequences, we use their variant Gated Recurrent Unit (GRU), which addresses this problem. GRU is mathematically defined as follows:

$$\mathbf{z}_t = \sigma_g(\hat{\mathbf{c}}_t W_z + \mathbf{h}_{t-1} U_z + b_z) \tag{4}$$

$$\mathbf{r}_t = \sigma_g(\hat{\mathbf{c}}_t W_r + \mathbf{h}_{t-1} U_r + b_r) \tag{5}$$

$$\bar{\mathbf{h}}_t = \phi_h(\hat{\mathbf{c}}_t W_h + (\mathbf{r}_t \odot \mathbf{h}_{t-1}) U_h + b_h) \tag{6}$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \bar{\mathbf{h}}_t \tag{7}$$

where $\mathbf{h}_{t-1}$ with $(t-1) \geq 0$ is the hidden state of the medical code embedding $\hat{\mathbf{c}}_{t-1}$; $\mathbf{z}_t$ and $\mathbf{r}_t$ are the update and reset gates associated with the medical code embedding $\hat{\mathbf{c}}_t$, respectively; $\bar{\mathbf{h}}_t$ is the hidden intermediate state; $\mathbf{h}_t$ is the hidden state of the current input $\hat{\mathbf{c}}_t$ and also the GRU's output; W_z, W_r, W_h, U_z, U_r, U_h, b_z, b_r; and b_h are GRU training parameters; and $\odot$ denotes the Hadamard product as usual.

The GRU's Eqs. (4)–(7) assume that the elapsed time δt between the recording dates of two consecutive diagnoses is regular. This assumption is not valid since δt may vary. In addition, this variation might be high. It is then crucial to weight each hidden state $\mathbf{h}_t$ of $\hat{\mathbf{c}}_t$ according to δt so that the model places more importance on recent diagnoses than those made a long time ago. Therefore, we introduce a decay factor in the GRU and multiply it by $\mathbf{h}_t$ to obtain a new hidden state $\tilde{\mathbf{h}}_t$. A GRU with a decay factor applied to its hidden state is called GRU-decay. Except for the hidden state, it has the same structure as a GRU. $\tilde{\mathbf{h}}_t$ is obtained as follows:

$$\tilde{\mathbf{h}}_t = exp(-max(0, \delta_t \tilde{W} + \tilde{b})) \odot \mathbf{h}_t \tag{8}$$

where $\tilde{W}$ and $\tilde{b}$ are learnable parameters. $\tilde{\mathbf{h}}_{t=T} = \tilde{\mathbf{h}}$ can be interpreted as a latent summary of the patient's health status.

As patient demographics (gender and age) are important factors to study for depression, we formally describe, in Section 3.4, how they are combined with the latent summary of patient health status $\tilde{\mathbf{h}}$ and then run through the classifier to predict whether the current patient will suffer from depression.

3.4. Depression detection: Classifier

To calculate the likelihood that a patient will be detected as a depressed patient, we first extract information from patient demographics via a *feedforward neural network* (FNN) and combine the extracted information with $\hat{\mathbf{h}}_t$. In end-to-end fashion, the result of this combination is fed into a set of stacked FNNs that play the role of the classifier. Formally, the classifier is

$$\hat{y} = f^1_{\beta_1} \circ f^2_{\beta_2} \circ \cdots \circ f^P_{\beta_P}(\langle \tilde{\mathbf{h}}, \tilde{\mathbf{d}} \rangle) \tag{9}$$

$$\tilde{\mathbf{d}} = g_\alpha(\mathbf{d}) \tag{10}$$

where g_α is an FNN with $Relu$ as an activation function; α is a set of learnable parameters of g; $\tilde{\mathbf{d}}$ is the latent representation of patient's demographics vector; $f^2_{\beta_2} \circ \cdots \circ f^P_{\beta_P}$ is P stacked FNNs with $Relu$ as an activation function; $\beta_2, \ldots, \beta_P$ are learnable parameters; $f^1_{\beta_1}$ is the final FNN with *sigmoid* as an activation function and β_1 as its learnable parameters; and $\hat{y} \in [0,1]$ is the likelihood that a patient suffers from depression.

We used a weighted binary cross-entropy as a loss function [81] to adjust the models' parameters while dealing with the problem of imbalanced classes. It is defined as follows:

$$\mathcal{L}_{wbc} = -\frac{1}{N} \sum_{n=1}^{N} \Big(w_1 * y_n * \ln(\hat{y}_n) + w_0 * (1 - y_n) * \ln(1 - \hat{y}_n) \Big) \tag{11}$$

where $w_0 = 1$ and $w_1 = N_0/N_1$ are the weighted factors of class 0 and 1, respectively. w_1 allows penalizing the model more when the class 1 is misclassified. Indeed, this choice is justified because we are dealing with imbalanced classes, i.e. the number of patients suffering from depression is much lower than those who do not suffer from it.

4. Experimentation

4.1. Settings

We coded the proposed model using Python 3.0 and the machine learning libraries Keras 2.4.3 and TensorFlow 2.4.0. All remaining pre-processing and performance evaluation was done with the libraries NumPy, Pandas and Scikit-learn. Finally, we ran the code on a cluster node with the following characteristics: An AMD Threadripper 3960X processor with 24 cores and 48 threads, 128 GB of memory, and an NVidia 3090 GPU with 24 GB of graphics memory.

4.2. Data

Our dataset was queried from the EHIF data warehouse and includes information on gender, birth year, ICD-10 coded primary and secondary diagnoses and the date of the treatment bill (diagnosis date) from 812,853 people (15 years or above) with a total of 26,973,943 diagnoses between 2018 and 2022. The data consist of all publicly insured people in Estonia with a depression diagnosis[8] (80,243 patients with 4,252,213 diagnoses). The control group consists of 732,610 patients (with 22,721,730 diagnoses), of which 498,764 people (with 10,779,835 diagnoses) did not have a psychiatric disorder diagnosed and 233,846 patients (with 11,941,895 diagnoses) had a psychiatric disorder other than depression. The percentage of insured people in Estonia is above 93.63% [82], so we are confident that our dataset is representative of the entire population.

Diagnoses were coded based on ICD-10. Each ICD-10 code consists of an alpha character known as a chapter, two digits describing the disease category, a dot and additional digits representing more details like the cause, location, severity or other clinical information (sub-categories). For example, F32.2 is the code for major depressive disorder, single episode, severe without psychotic features. F stands for mental and behavioural disorders; F30–F39 are codes for mood [affective] disorders; and F32 is the category of major depressive disorder, single episode. The '.2' at the end of F32.2 specifies the severity. All patients with the ICD-10 codes F33.x or F32.x are classified as patients with depression. For the latter, only diagnoses made before being diagnosed with depression are taken into account in the study. Those which followed the depression diagnosis are ignored. Fig. 2 shows the data extraction process.

The Research Ethics Committee of the National Institute for Health Development (TAIEK[9]) approved this study's research design and data usage (Decision No. 1148).

---

[8] To overcome potential data leakage, we considered all diagnoses starting with F32 (major depressive disorder) and F33 (recurrent depressive disorder) as 'depression'.

[9] Tervise Arengu Instituudi inimuuringute eetikakomitee.

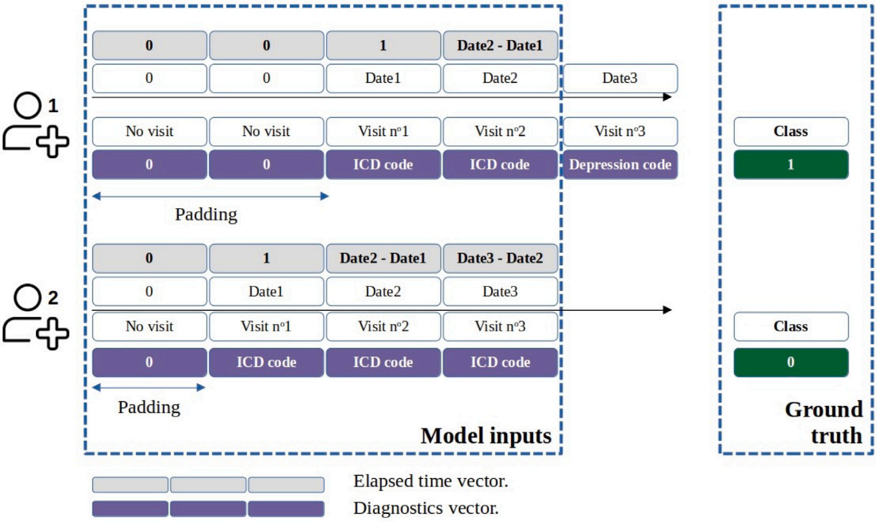


Fig. 2. Data extraction process from two patients. For padding values, we assign 0 as elapsed time. For the first diagnosis (ICD code of the first visit), we assign 1 as the elapsed time. When a depression code is observed in the diagnosis list, the associated ground truth is 1. All diagnoses after the first depression diagnosis are ignored. On the other hand, when no depression code is observed, the ground truth associated with the sample is 0.

4.3. Model and training hyperparameters

We performed an extensive grid search over embedding_space = $\{50, 80, 100\}$, dimenssion_linear_space = $\{32, 64, 80, 100\}$, GRU_decay_units = $\{30, 50, 80, 100\}$, demographics_FNN_units = $\{10, 20, 30\}$, classifier_FNN_units = $\{20, 30, 50\}$, number_epochs = $\{20, 30, 40, 50, 60, 70, 80\}$, optimizer = $\{Adam, SGD, RMSprop\}$ to find the optimal value for each hyperparameter of the model. The values retained for each hyperparameter are as follows:

- The dimension of the embedding space was set to 50.
- We used a mask on the embedding layer to skip padding values during the calculation.
- For the attention layer, we set the dimension of linear spaces at 80.
- The number of GRU-decay units was set to 50.
- We applied a dropout of 0.5 on the hidden layer of GRU-decay to prevent gradient problems.
- Concerning the FNN dedicated to the extraction of demographic features of patients, we defined the number of units as 10.
- The classifier comprises two stacked FNNs, each with 20 and 1 units, respectively.

Once more, using the grid search technique, we defined the training hyperparameters as follows:

- The number of epochs was set to 20.
- The batch size was set to 1500.
- We used $Adam$ as the optimizer.
- The learning rate was set to 0.001.

Table 1 summarizes all the hyperparameter values.

Table 1
Model and training hyperparameters.

Hyperparameters	Values
Dimension of the embedding space	50
Dimension of linear spaces	80
Number of GRU-decay units	50
GRU-decay dropout	0.5
Number of FNN units of patient's demographics	0.5
Number of FNN units of the classifier	20 & 1
Number of epochs	20
Batch size value	1500
Optimizer	$Adam$

4.4. Results

To assess the performance of our model, we use the *area under the ROC Curve* (AUC) and the *area under the precision–recall curve* (AUPRC) as metrics. The *precision–recall* curve is a function of *recall* (12) on the x-axis and *precision* (13) on the y-axis. The *receiver operating characteristic* (ROC) curve is a function of *false positive rate* (14) on the x-axis and *recall* on the y-axis.

$$Recall = Sensitivity = \frac{|TP|}{|TP| + |FN|} \tag{12}$$

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{13}$$

$$FalsePositiveRate = \frac{|FP|}{|FP| + |TN|} \tag{14}$$

where $|TP|$ is the number of true positives, $|FN|$ the number of false negatives, $|TN|$ the number of true negatives and $|FP|$ the number of false positives. Indeed, by varying the threshold when calculating recall, precision and the false positive rate, these metrics avoid biased scores caused by the high number of non-target classes, i.e. the class 0. They are suitable for assessing the performance of models in the face of an imbalanced class problem.

We compare the average AUC and AUPRC scores obtained over 5-fold cross-validation with those of the following models: logistic regression (LR); feedforward neural network (FNN); long short-term memory (LSTM); convolutional neural network combined with LSTM (CNN-LSTM); gated recurrent unit with a decay factor (GRU-decay);

Table 2
AUC and AUPRC scores on depression detection task over 5-cross validation. ± denotes the standard deviation.

Models	AUC	AUPRC
LR	0.813 ± 0.002	0.296 ± 0.003
CNN-LSTM	0.849 ± 0.002	0.394 ± 0.009
LSTM	0.848 ± 0.001	0.385 ± 0.005
FNN	0.837 ± 0.002	0.374 ± 0.006
GRU-decay	0.989 ± 0.001	0.972 ± 0.001
GRU-Δt	0.986 ± 0.002	0.961 ± 0.005
Att-GRU-decay	**0.990 ± 0.001**	**0.974 ± 0.002**

Table 3
Specificity and sensitivity scores on the depression detection task.

Models	Specificity		Sensitivity		Time (min)	
	Threshold				Train	Test
	0.5	0.8	0.5	0.8		
LR	0.705	0.960	0.787	0.263	8.680	**0.008**
CNN-LSTM	0.718	0.902	0.818	0.549	8.299	0.035
LSTM	0.724	0.916	0.814	0.523	30.513	0.050
FNN	0.714	0.911	0.810	0.528	**1.312**	0.006
GRU-decay	**0.995**	**0.999**	0.939	0.926	49.547	0.068
GRU-Δt	0.962	0.987	0.962	0.936	2.877	0.102
Att-GRU-decay	0.985	**0.999**	**0.955**	**0.944**	56.754	0.102

and a gated recurrent unit taking as inputs diagnostic vectors concatenated to the elapsed time vectors (GRU-Δt). All results are reported in Table 2.

From Table 2, we can clearly see that our proposed model achieves the best performances. Although the GRU-decay and GRU-Δt results are very accurate, ours are slightly better. Compared to our Att-GRU-decay model and the GRU-decay model, GRU-Δt is less accurate because it does not incorporate any explicit techniques to better learn existing patterns between diagnoses and time. The slight superiority of our model highlights the additional contribution of the self-attention layer in the decision-making process. Indeed, unlike the GRU-decay, which only benefits from decay factors that prevent the prediction from being based on diagnoses made a long time ago, our model, thanks to the self-attention mechanism, will also detect hidden patterns existing between diagnoses that may be the cause of possible depression in the patient. Where the difference between the AUC scores of the models is not so large, the AUPRC scores of our model and the GRU-decay model far exceed those of the other competitors. This huge difference reveals how crucial it is to weigh the significance of the diagnoses according to their respective recording dates.

It is not surprising that the LR model, which is a traditional machine learning model, performs worse than the other models, which are deep learning models. Indeed, unlike machine learning, which is somewhat dependent on feature engineering, deep learning can extract hidden features by itself thanks to its non-linear functions and therefore does not need feature engineering. This property makes deep learning models more accurate than machine learning models when processing data with complex patterns. Machine learning models can sometimes achieve results similar to or better than deep learning models [83]. Moreover, they are more explainable. Another aspect that reveals the results in Table 2 is the low accuracy of non-sequential models such as LR and FNN compared to others designed for sequence modelling. Thus, we conclude that processing patients' diagnoses at different dates with non-sequential models leads to losing temporal patterns in depression detection. Compared to GRU-decay and our model, CNN-LSTM and LSTM, also models designed to handle sequential data, failed because they processed diagnoses as if they were made at regular time intervals.

As AUC and AUPRC are calculated from different thresholds, we also investigate the specificity (15) and the sensitivity of the models on fixed threshold values of 0.5 and 0.8. This second evaluation was carried out on a single loop of the 5-cross validation.

$$Specificity = \frac{|TN|}{|TN| + |FP|} \tag{15}$$

Indeed, the higher the threshold, the more confidence practitioners have in the model's outcome. A higher threshold is even more important in the medical field, as misdiagnoses can have irreversible consequences. The specificity and sensitivity scores of all models calculated from the confusion matrices in Fig. 3 are reported in Table 3. We also report the training and testing time for each model to give an idea of how long it will take for each of them to produce results in a real deployment. The ROC curves and precision–recall curves of the evaluated models are shown in Fig. 4

Table 3 and Fig. 4 show that GRU-decay, GRU-Δt and ours obtain the best specificity scores, sensitivity scores, ROC curves and Precision_Recall curves. These scores again show how incorporating a decay factor to handle better diagnoses recorded at different dates improves the classification task. We note that almost all models provide satisfactory results with a threshold set to 0.5. We assume that these results are due to the large qualitative amount of data and the weighted binary cross-entropy, which improves the models' ability to classify the minority class, i.e. the depressed patient. If a threshold is set to 0.5, the sensitivity scores for all models are fairly accurate. We find a considerable drop in the performance of the LR, CNN-LSTM, LSTM and FNN models when the threshold is set to 0.8. On the other hand, the GRU-decay model and ours remain very accurate. Although the specificity score of the GRU-decay model with a threshold of 0.5 is better than that obtained with ours, with the other configurations, our model is better overall. It is worth mentioning that, despite the high threshold value of 0.8, we obtained spectacular sensitivity and specificity scores close to 1. In verbal form, among the 146,522 non-depressed patients in the training set, our model correctly classifies 146,409 with a probability of 0.8%. For the 13,677 depressed patients, our model correctly classifies 12,906 with a probability of 0.8%.

For classification problems such as those related to medicine, the output of the models must be very accurate to avoid misdiagnoses leading to inappropriate treatment. Especially for the early detection of psychiatric diseases such as depression, the model's sensitivity is crucial. With the quantitative results we have obtained, we are very confident that our model can help medical professionals in their decision-making to detect patients with depression faster and thus significantly reduce the misdiagnosis rate.

We note that in terms of training and testing times, our model takes the longest. This is partly due to the number of parameters (97,121) and the time complexity of the self-attention and GRU-decay mechanisms. Despite having the longest test duration, 0.102 min is still sufficient for using it in the clinical process. The number of parameters in the competing models is shown in Appendix A.6.

In the next section, we conduct different ablation studies to show how the model works in different configurations.

5. Ablation studies

We have devoted this section to evaluating the model in the following configuration: (i) without the decay factor; and (ii) with and without patient demographics.

Without the decay factor. In Table 4, we observed a considerable drop in performance when the decay factor is not taken into account. These results support our assertion regarding the importance of accounting for irregular elapsed time between visits. Indeed, the normal GRU fails because it processes diagnoses as if they were made at regular intervals and is therefore unable to capture the correct underlying temporal pattern of diseases

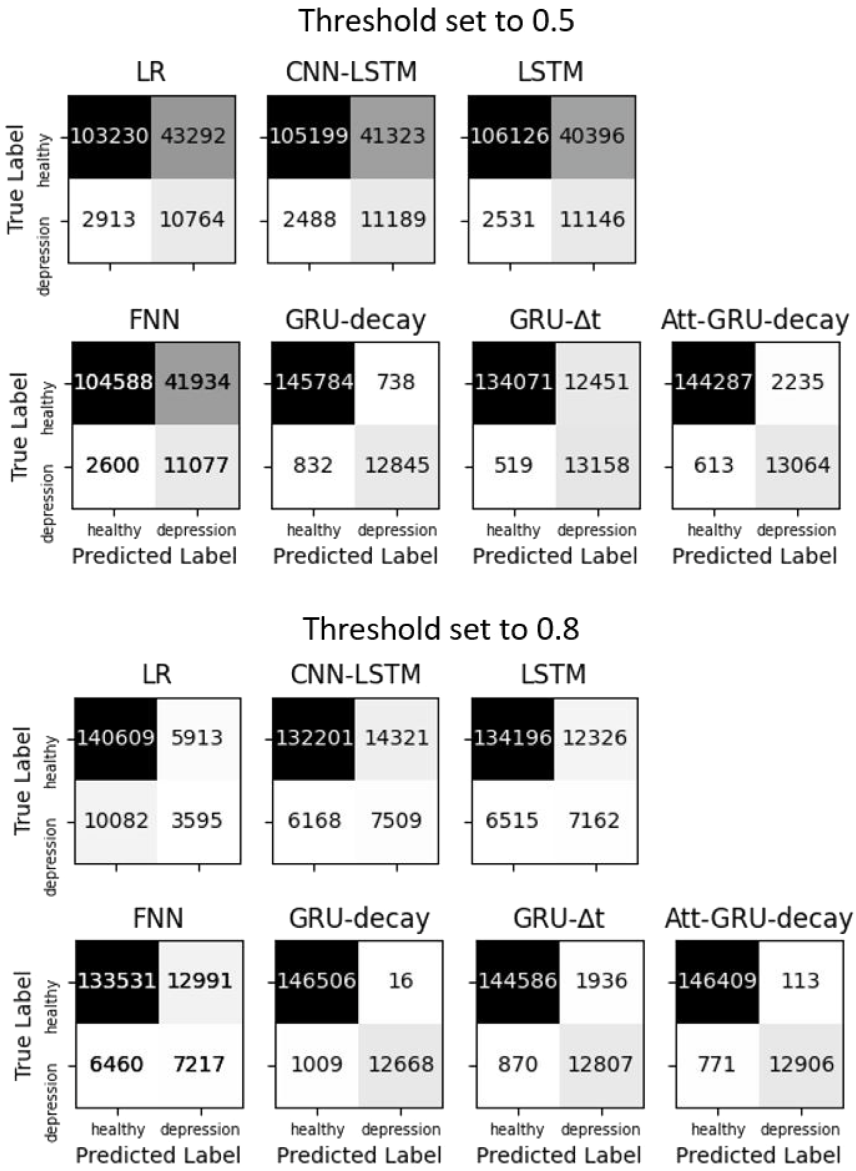


Fig. 3. Confusion matrices.

Table 4
Evaluation of the model without the decay factor over 5-cross validation.

Models	AUC	AUPRC
Att-GRU	**0.853 ± 0.001**	**0.405 ± 0.007**
Att-GRU-decay	**0.990 ± 0.001**	**0.974 ± 0.002**

With and without patient demographics. The AUC and AUPRC scores in Table 5 show that patient demographics have little influence on the detection of depression. We can see that without patient demographics, the performance of the model is not affected. However, when patient demographics are used exclusively, model performance drops significantly. We conclude that the model can still produce accurate results when patient demographics are not available Scientific literature suggests an impact of demographic factors like gender [84] or age [85] on the likelihood of getting depression. We assume that this effect is not visible in our ablation study because the model learns gender and age trends through associated diseases.

As quantitative results are not sufficient to guarantee the veracity of a model in medical applications, we also propose a component for extracting disease patterns that influence the model output to be able to give a qualitative interpretation of the model behaviour (see Section 6).

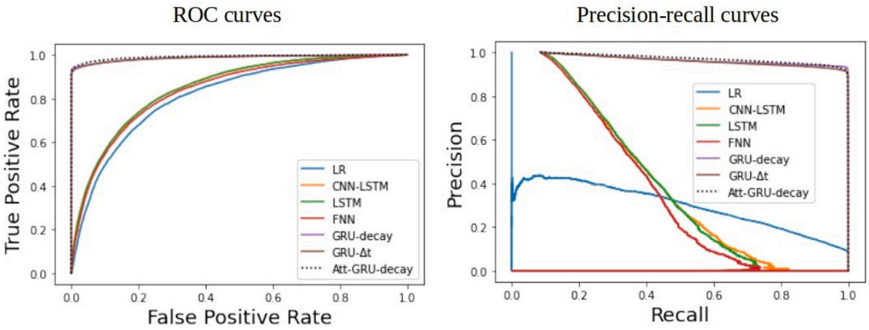


Fig. 4. ROC and Precision–recall curves.

Table 5
Evaluation of the model without patient demographics and exclusively with over 5-cross validation. ex/pd stands for exclusively with patient demographics, and wo/pd stands for without patient demographics.

Models	AUC	AUPRC
Att-GRU-decay ex/pd	**0.647** ± **0.002**	**0.131** ± **0.002**
Att-GRU-decay wo/pd	**0.990** ± **0.001**	**0.972** ± **0.001**
Att-GRU-decay	**0.990** ± **0.001**	**0.974** ± **0.002**

6. Uncovering disease patterns

The following section shows the interaction between features in the attention layer of our model.

Apart from the benefits of increased prediction accuracy, we use self-attention to provide insights into the disease relationships the model has learned. We propose using this to give medical professionals a better understanding of the model by showing that it can correctly identify commonly known disease correlations. Those disease correlations can also be used to infer rules and find indicator diseases [57].

The alignment matrix in Fig. 5 shows a given patient's last seven ICD-10 codes on the x-axis and how our trained neural network associates them with each other. The colour indicates the strength of the correlation, from blue (not correlated) to red (strongly correlated). In this example, our trained network identified a strong correlation between heart failure and type 2 diabetes. This correlation is already well known in medicine and shows how the Att-GRUD-decay could infer it from the training data.

Now, consider the second patient (Fig. 6). We see the last ten diagnoses, from which the model identified that oesophagitis is correlated with migraine, dorsalgia and abdominal and pelvic pain. While abdominal and pelvic pain could logically make sense, there is currently no strong medical evidence for a correlation with migraine or dorsalgia. Nevertheless, some forms of migraine trigger strong nausea, which could lead to oesophagitis and spinal problems, manifesting as dorsalgia and negatively influencing migraines.

The third patient (Fig. 7) shows a strong correlation between the need for immunization against other single viral diseases and sleep disorders and retinal disorders. We are unaware of any medical evidence of a correlation between those ICD-10 codes.

This example demonstrates how the model learned and can find reasonable connections from large data sets. Still, not all correlations are evidence-based from a medical perspective.

Generally, the correlations found can be sorted into three categories:

1. *True correlations*, which are (based on our current medical knowledge) reasonable (Fig. 5, and potentially Fig. 6).
2. *"Hallucinations"* of the deep learning network, i.e. output that does not seem to be justifiable based on the training data (potentially Figs. 6 and 7).
3. *Potentially true correlations*, which we currently cannot grasp because they exceed today's medical knowledge (Fig. 6, potentially Fig. 7).

So, while the prediction accuracy of our model is high, the individual correlations shown by the self-attention still need to be evaluated carefully.

It is important to note that these correlations are only based on the attention layer of our model. They do not offer explainability of other parts, e.g. the GRU component, of our model.

7. Discussion

Several high-performing AI models have already been proposed in the healthcare sector [86–89]. Still, success stories of AI providing real clinical value are rare. The reasons for this include a lack of data availability, integration into clinical processes and lack of trust due to the black-box characteristics of the models. In the previous sections, we demonstrated that our novel Att-GRU-decay model outperforms the current state of the art. In this section, we elaborate on a possible application scenario to demonstrate how this model could improve the status quo while avoiding the above-mentioned pitfalls.

Since one of the main problems in psychiatry is that patients with psychiatric disorders are often diagnosed late, we propose to use this model to screen patients when they visit a healthcare professional proactively. This makes sense, especially for general practitioners (GPs) with high patient turnover. The model can be plugged into the GP's systems and rolled out at the insurance provider level or on a national level on top of an NHIS via a RESTful [90] API. If the GP enters the diagnosis at the end of the visit, our model enables the doctor's IT systems to send an alert if the patient is thought to have undiagnosed depression. The GP can then re-evaluate the decision, using our explainability component, and refer the patient to a specialist for further treatment in the case of a true positive prediction. Since the proposed system operates on diagnoses from medical claims data, which medical professionals capture during their work anyway, no additional effort is needed. This allows seamless integration into the current clinical workflow. Because of the high specificity, we assume the risk of alert fatigue is low. On the other hand, if we compare our sensitivity of 94.4% to the reported 50.1% (95% CI: 41.3 to 59.0) sensitivity of GPs for diagnosing depression [4], we see that our model has the potential to decrease the number of undiagnosed depression patients significantly. It even outperforms population-level screening questionnaires, such as the PHQ-9, which has a sensitivity of 88% and a specificity of 88% [91]. In addition, time-wise, the suggested approach outperforms the current use of questionnaires and assessment

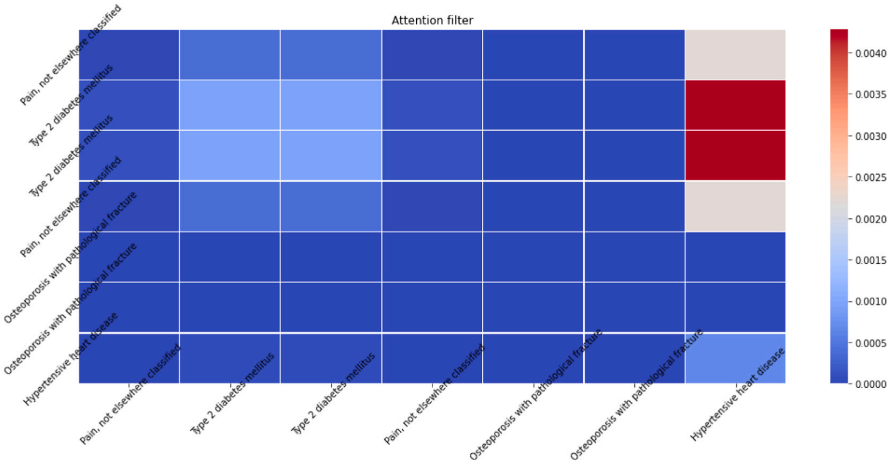


Fig. 5. Attention filter — Example 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

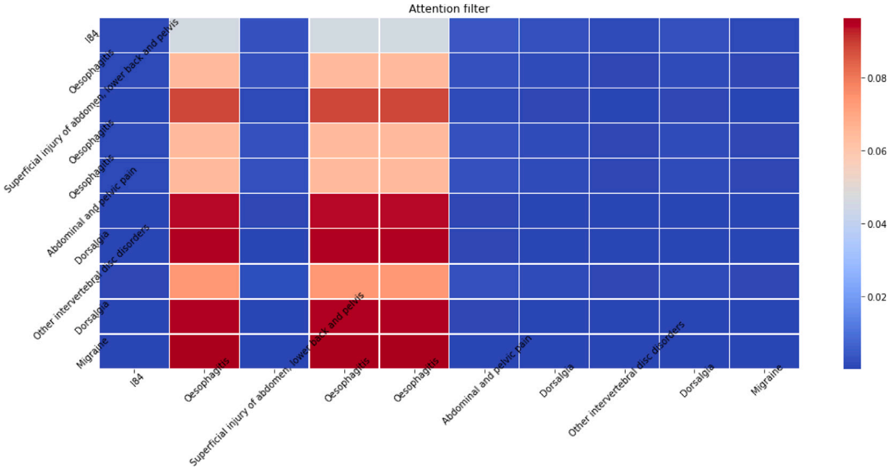


Fig. 6. Attention filter — Example 2.

scales. Current depression assessment instruments, such as the Beck Depression Inventory [9], the Hamilton Depression Rating Scale [10] or the Montgomery-Åsberg Depression Rating Scale [92], take between 15 and 30 min to complete. At the same time, our proposed screening approach outputs results in seconds.

This use case can be expanded to screening other diseases in domains other than mental health as long as the data utilized have the same structure. Since we operate based on medical claims data available in most countries and cover a wide range of medical information, it should be fairly easy to retrain our model to predict other diseases.

It is important to stress that we are not proposing to replace medical doctors with AI algorithms. We suggest that AI algorithms can be used as screening instruments, assisting doctors by discovering hidden patterns in large volumes of medical data to help them diagnose faster and more accurately. The output of an AI model still needs to be validated, checked against the current patient situation, and communicated. Furthermore, the subsequent steps, i.e. further diagnostic procedures and treatment decisions, still need to be taken by doctors.

We are fairly confident that the model will perform well in a production setting because of the large amount of real-world data used for training and evaluation, which includes nearly every adult Estonian. For further research, the model needs to be evaluated in a randomized control trial (RCT) to obtain further evidence on its usefulness in a clinical setting. One limitation of our study is that the data we used as ground truth might be biased, for instance, because of the previously described low accuracy of human diagnoses, but also because medical claims data are used for billing purposes, which creates an incentive for medical professionals to adapt codes to maximize revenue. An RCT can help show the impact of this potential bias on the usefulness of our proposed model. Another limitation of our research is that we did not use any prescription, laboratory, genomics data or other unobtrusive data sources. We focused solely on diagnostic and socio-demographic data because this is easily accessible during the clinical process without the need for any specific data collection by the physician or patient. Because of the good results of our approach, we saw the exploration of other data sources as out of scope. Nevertheless, we encourage further research to analyse whether other AI algorithms based on other clinical data sources can give similar or better results. Additionally, we encourage further research to evaluate the described scenario with other digital health evaluation methods to assess usability and efficacy.

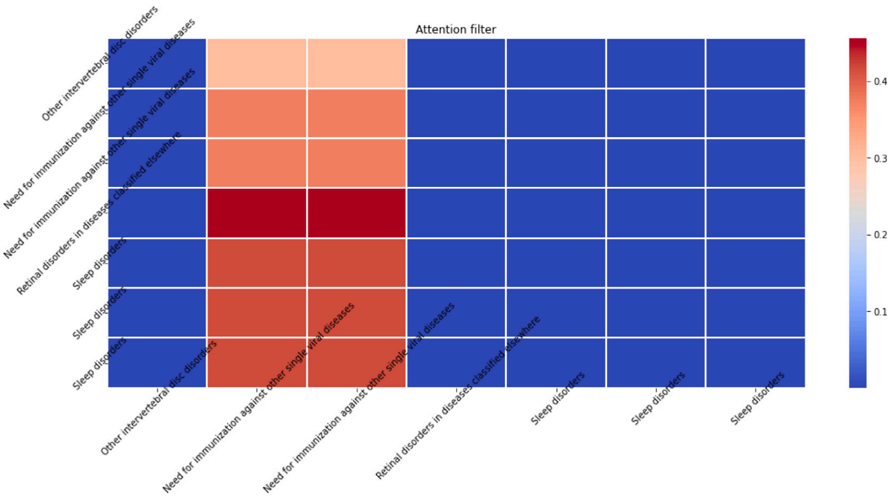


Fig. 7. Attention filter — Example 3.

Further research is also planned to investigate how well the model can be applied to different diseases using the same kind of data. We see the use of self-attention rather than multi-head attention as a potential limitation in terms of explainability and disease correlations. The use of multi-head attention could potentially find more and deeper hidden disease patterns.

8. Conclusion

In this research, we used the medical claims data of 812,853 patients with 26,973,943 diagnoses to evaluate deep learning for depression detection. We contribute by evaluating the most common deep learning algorithms and introducing our novel Att-GRU-decay model, which outperforms other state-of-the-art deep learning models with an AUC of 0.99 and an AUPRC of 0.974. We further describe a potential application scenario for using the proposed model for screening patients in a GP setting. Since the use of real-world data covers nearly every adult Estonian, the excellent accuracy results of Att-GRU-decay, in addition to the proposed use-case scenario with a potential increase in the specificity of depression diagnosis by GPs from 50.1% to as much as 94.4%, we see this research as a potential game changer for psychiatric screening.

CRediT authorship contribution statement

Markus Bertl: Conceptualization, Data curation, Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Nzamba Bignoumba:** Conceptualization, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Peeter Ross:** Funding acquisition, Supervision, Writing – review & editing. **Sadok Ben Yahia:** Supervision, Writing – review & editing. **Dirk Draheim:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We want to thank the EHIF for providing access to the data needed for this research. The Estonian Human Research Ethics Committee (TAIEK) of the Institute for Health Development (Decision No. 1148) approved this study's research design and data usage.

Appendix A. Number of parameters per model

See Table A.6.

Table A.6
Number of parameters per model.

Models	# of parameters
LR	1,649
CNN-LSTM	136,849
LSTM	101,871
FNN	5,156,331
GRU-decay	96,725
GRU-Δt	107,326
Att-GRU-decay	97,121

Appendix B. Data distribution

See Figs. B.8–B.12.

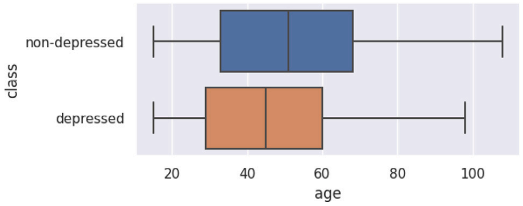


Fig. B.8. Age distribution.

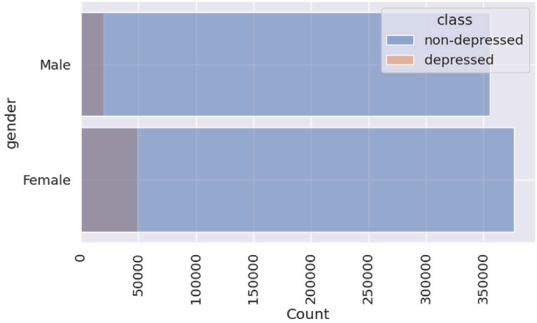


Fig. B.9. Gender distribution.

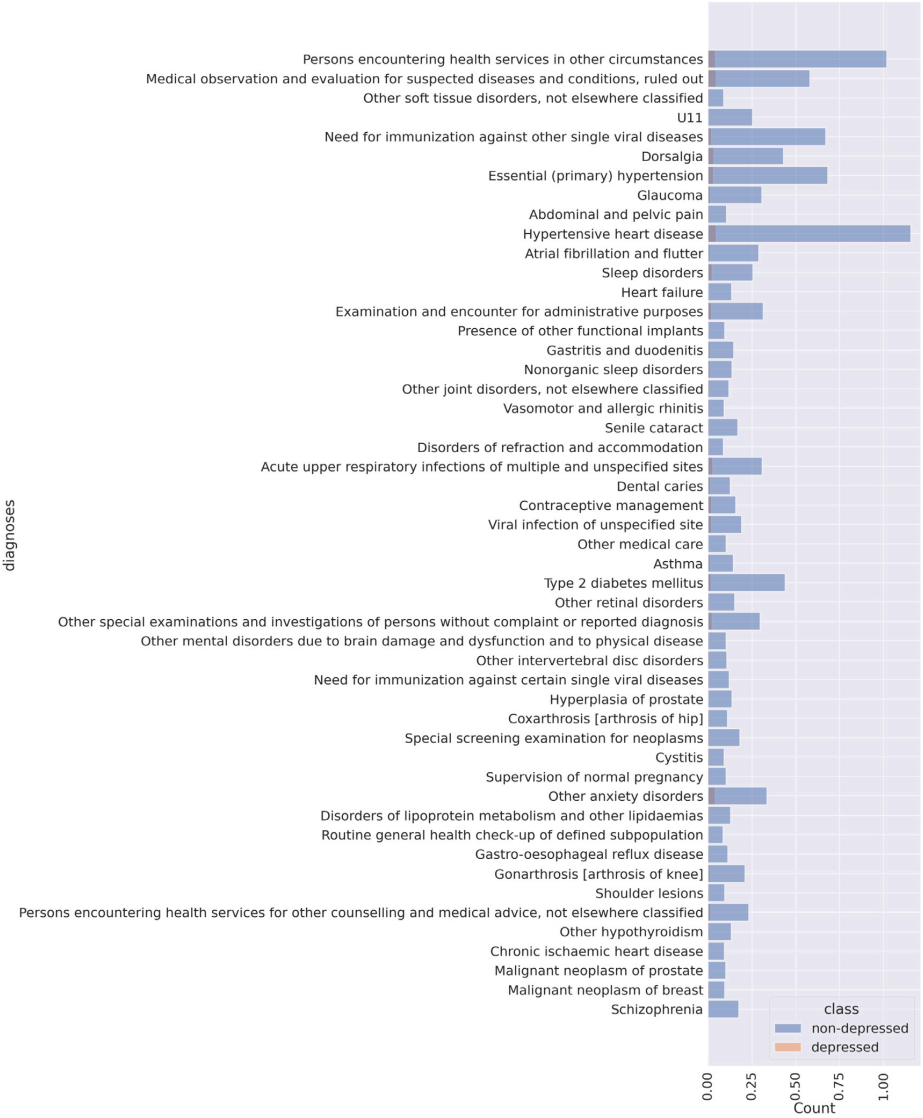


Fig. B.10. Top 50 ICD-10 codes in our dataset.

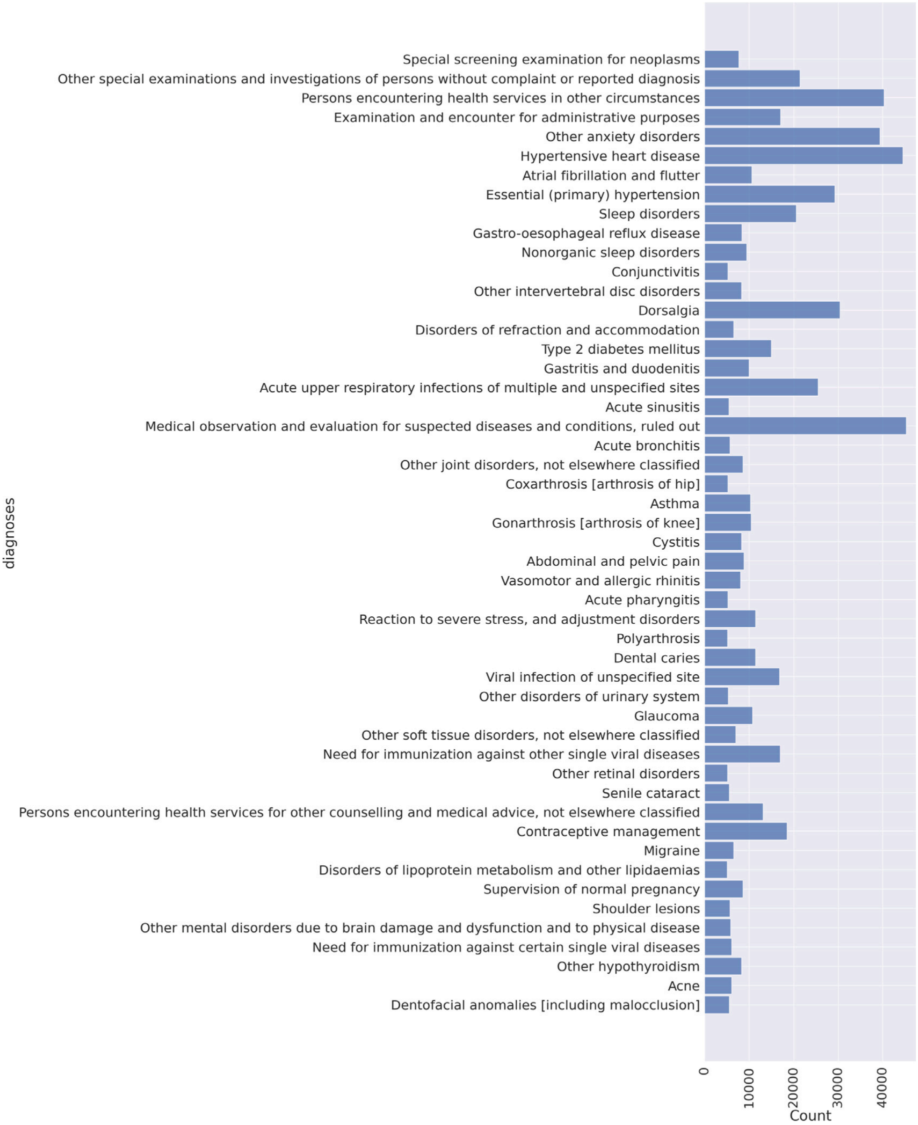


Fig. B.11. Top 50 ICD-10 codes for patients with depression.

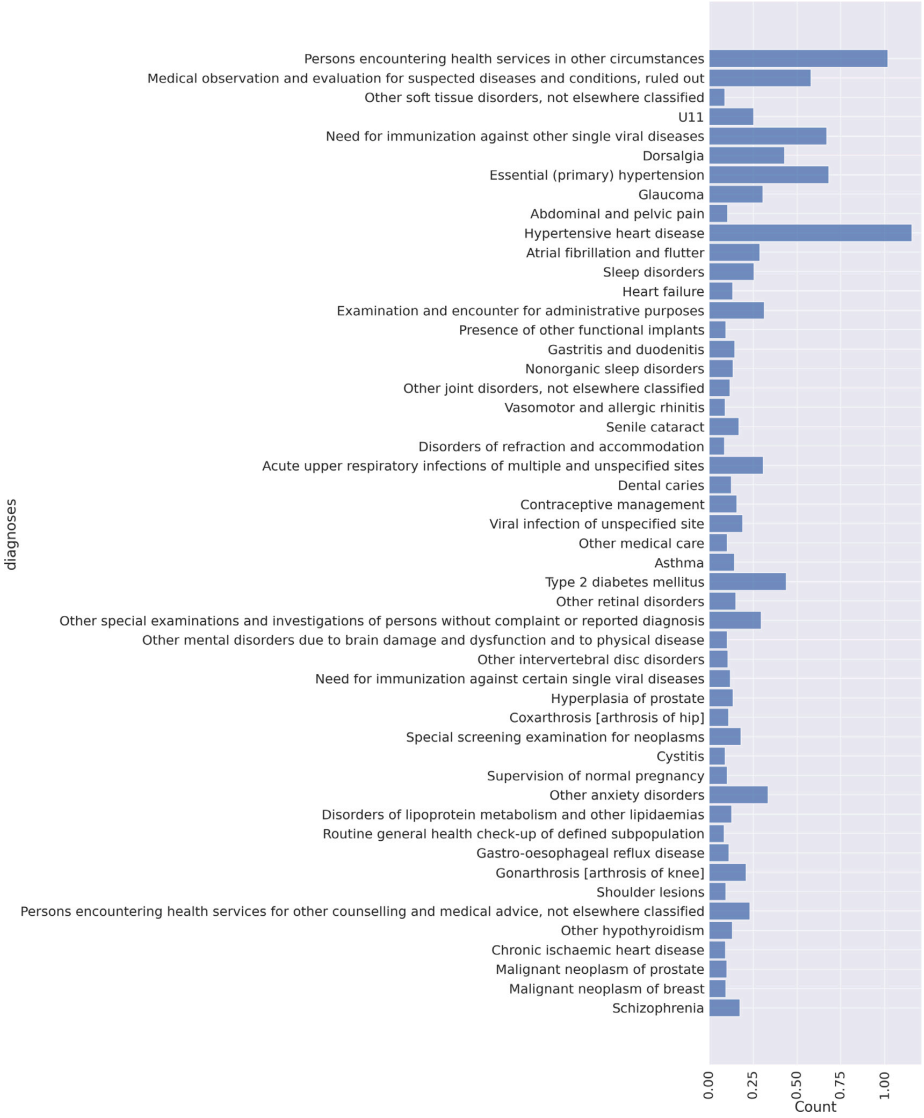


Fig. B.12. Top 50 ICD-10 codes for patients without depression.

References

[1] Wittchen H-U, Jacobi F, Rehm J, Gustavsson A, Svensson M, Jönsson B, Olesen J, Allgulander C, Alonso J, Faravelli C, et al. The size and burden of mental disorders and other disorders of the brain in Europe 2010. Eur Neuropsychopharmacol 2011;21(9):655–79.

[2] McGlynn EA, Asch SM, Adams J, Keesey J, Hicks J, DeCristofaro A, Kerr EA. The quality of health care delivered to adults in the United States. N Engl J Med 2003;348(26):2635–45.

[3] Aboraya A, Rankin E, France C, El-Missiry A, John C. The reliability of psychiatric diagnosis revisited: The clinician's guide to improve the reliability of psychiatric diagnosis. Psychiatry (Edgmont) 2006;3(1):41.

[4] Mitchell AJ, Vaze A, Rao S. Clinical diagnosis of depression in primary care: a meta-analysis. Lancet 2009;374(9690):609–19.

[5] Wamala SP, Lynch J, Horsten M, Mittleman MA, Schenck-Gustafsson K, Orth-Gomer K. Education and the metabolic syndrome in women. Diabetes Care 1999;22(12):1999–2003.

[6] Gustavsson A, Svensson M, Jacobi F, Allgulander C, Alonso J, Beghi E, Dodel R, Ekman M, Faravelli C, Fratiglioni L, et al. Cost of disorders of the brain in Europe 2010. Eur Neuropsychopharmacol 2011;21(10):718–79.

[7] Williams SZ, Chung GS, Muennig PA. Undiagnosed depression: A community diagnosis. SSM-Population Health 2017;3:633–8.

[8] American Psychology Association. Depression assessment instruments. 2023, URL https://www.apa.org/depression-guideline/assessment.

[9] Beck AT, Ward CH, Mendelson M, Mock J, Erbaugh J. An inventory for measuring depression. Arch Gen Psychiatry 1961;4(6):561–71.

[10] Hamilton M. A rating scale for depression. J Neurol Neurosurg Psychiatry 1960;23(1):56.

[11] Bertl M, Ross P, Draheim D. A survey on AI and decision support systems in Psychiatry – Uncovering a Dilemma. Expert Syst Appl 2022.

[12] Bertl M, Metsallik J, Ross P. Digital Decision Support Systems for Post-Traumatic Stress Disorder – Implementing a novel framework for decision support systems based on a technology-focused, systematic literature review. Front Psychiatry 2022. http://dx.doi.org/10.3389/fpsyt.2022.923613.

[13] Bertl M, Ross P, Draheim D. Systematic AI support for decision-making in the healthcare sector: Obstacles and success factors. Health Policy Technol 2023;100748.

[14] Bertl M, Klementi T, Piho G, Ross P, Draheim D. How domain engineering can help to raise adoption rates of artificial intelligence in healthcare. In: Information integration and web intelligence. Springer Nature Switzerland; 2023, p. 3–12.

[15] Bertl M, Kankainen KJI, Piho G, Draheim D, Ross P. Evaluation of Data Quality in the Estonia National Health Information System for Digital Decision Support. In: Proceedings of the international health data workshop. 2023.

[16] Ivers N, Brown AD, Detsky AS. Lessons from the Canadian experience with single-payer health insurance: Just comfortable enough with the status quo. JAMA Int Med 2018;178(9):1250–5. http://dx.doi.org/10.1001/jamainternmed.2018.3568.

[17] Wendt C. Changing healthcare system types. Soc Policy Adm 2014;48(7):864–82.

[18] Grosios K, Gahan PB, Burbidge J. Overview of healthcare in the UK. EPMA J 2010;1(4):529–34.

[19] McBride K, Toots M, Kalvet T, Krimmer R. Leader in e-government, Laggard in open data: Exploring the case of Estonia. In: Revue francaise d'administration publique, vol. 3. École nationale d'administration; 2018, p. 613–25.

[20] Lips S, Tsap V, Bharosa N, Krimmer R, Tammet T, Draheim D. Management of national eID infrastructure as a state-critical asset and public-private partnership: Learning from the case of estonia. Inf Syst Front 2023. http://dx.doi.org/10.1007/s10796-022-10363-5.

[21] Metsallik J, Ross P, Draheim D, Piho G. Ten years of the e-health system in Estonia. In: CEUR workshop proceedings, vol. 2336. 2018, p. 6–15.

[22] Estonian National Electoral Committee and the State Electoral Office. Valimised – Voting results in detail. 2019, URL https://ep2019.valimised.ee/en/voting-result/index.html, Last accessed on 2022-03-04.

[23] Parv L, Kruus P, Motte K, Ross P. An evaluation of e-prescribing at a national level. Inf Health Soc Care 2016;41(1):78–95.

[24] Conneau A, Schwenk H, Barrault L, Lecun Y. Very deep convolutional networks for natural language processing 2016;2(1), arXiv preprint arXiv:1606.01781.

[25] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst 2017;30.

[26] Wu C-S, Chen C-H, Su C-H, Chien Y-L, Dai H-J, Chen H-H. Augmenting DSM-5 diagnostic criteria with self-attention-based BiLSTM models for psychiatric diagnosis. Artif Intell Med 2023;136:102488.

[27] Zhang Y-D, Dong Z, Wang S-H, Yu X, Yao X, Zhou Q, Hu H, Li M, Jiménez-Mesa C, Ramirez J, et al. Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. Inf Fusion 2020;64:149–87.

[28] Xu W, Su C, Li Y, Rogers S, Wang F, Chen K, Aseltine R. Improving suicide risk prediction via targeted data fusion: Proof of concept using medical claims data. J Am Med Inform Assoc 2022;29(3):500–11.

[29] Chen ZS, Galatzer-Levy IR, Bigio B, Nasca C, Zhang Y, et al. Modern views of machine learning for precision psychiatry. Patterns 2022;3(11):100602.

[30] Jing Y. Intelligent assessment of mental health based on multisource information fusion. J Healthc Eng 2022;2022.

[31] Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent neural networks for multivariate time series with missing values. Sci Rep 2018;8(1):1–12.

[32] Lee Y, Jun E, Suk H-I. Multi-view integration learning for irregularly-sampled clinical time series. 2021, arXiv preprint arXiv:2101.09986.

[33] Cao W, Wang D, Li J, Zhou H, Li L, Li Y. Brits: Bidirectional recurrent imputation for time series. Adv Neural Inf Process Syst 2018;31.

[34] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, arXiv preprint arXiv:1412.3555.

[35] Meng Y, Speier W, Ong MK, Arnold CW. Bidirectional representation learning from transformers using multimodal electronic health record data to predict depression. IEEE J Biomed Health Inf 2021;25(8):3121–9.

[36] Bertl M, Ross P, Draheim D. Predicting psychiatric diseases using AutoAI: A performance analysis based on health insurance billing data. In: International conference on database and expert systems applications. Springer; 2021, p. 104–11.

[37] Hosseinifard B, Moradi MH, Rostami R. Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from EEG signal. Comput Methods Programs Biomed 2013;109(3):339–45.

[38] Thandapani S, Mahaboob MI, Iwendi C, Selvaraj D, Dumka A, Rashid M, Mohan S. IoMT with deep CNN: AI-based intelligent support system for pandemic diseases. Electronics 2023;12(2):424.

[39] Sardari S, Nakisa B, Rastgoo MN, Eklund P. Audio based depression detection using Convolutional Autoencoder. Expert Syst Appl 2022;189:116076.

[40] Ma X, Yang H, Chen Q, Huang D, Wang Y. Depaudionet: An efficient deep model for audio based depression classification. In: Proceedings of AVEC'16 – the 6th international workshop on audio/visual emotion challenge. ACM; 2016, p. 35–42.

[41] Nasir M, Jati A, Shivakumar PG, Nallan Chakravarthula S, Georgiou P. Multi-modal and multiresolution depression detection from speech and facial landmark features. In: Proceedings of AVEC'16 – the 6th international workshop on audio/visual emotion challenge. ACM; 2016, p. 43–50.

[42] Ay B, Yildirim O, Talo M, Baloglu UB, Aydin G, Puthankattil SD, Acharya UR. Automated depression detection using deep representation and sequence learning with EEG signals. J Med Syst 2019;43:1–12.

[43] Liao S-C, Wu C-T, Huang H-C, Cheng W-T, Liu Y-H. Major depression detection from EEG signals using kernel eigen-filter-bank common spatial patterns. Sensors 2017;17(6):1385.

[44] Bachmann M, Päeske L, Kalev K, Aarma K, Lehtmets A, Ööpik P, Lass J, Hinrikus H. Methods for classifying depression in single channel EEG using linear and nonlinear signal analysis. Comput Methods Programs Biomed 2018;155:11–7.

[45] Avots E, Jermakovs K, Bachmann M, Päeske L, Ozcinar C, Anbarjafari G. Ensemble approach for detection of depression using EEG features. Entropy 2022;24(2):211.

[46] Wang R, Wang W, DaSilva A, Huckins JF, Kelley WM, Heatherton TF, Campbell AT. Tracking depression dynamics in college students using mobile phone and wearable sensing. Proc ACM Interact Mob Wearable Ubiquitous Technol 2018;2(1):1–26.

[47] Rykov Y, Thach T-Q, Bojic I, Christopoulos G, Car J, et al. Digital biomarkers for depression screening with wearable devices: cross-sectional study with machine learning modeling. JMIR mHealth uHealth 2021;9(10):e24872.

[48] Moshe I, Terhorst Y, Opoku Asare K, Sander LB, Ferreira D, Baumeister H, Mohr DC, Pulkki-Råback L. Predicting symptoms of depression and anxiety using smartphone and wearable data. Front Psychiatry 2021;12:625247.

[49] Coutts LV, Plans D, Brown AW, Collomosse J. Deep learning with wearable based heart rate variability for prediction of mental and general health. J Biomed Inform 2020;112:103610.

[50] Kipli K, Kouzani A, A Hamid IR. Investigating machine learning techniques for detection of depression using structural MRI volumetric features. Int J Biosci, Biochem Bioinform 2013;3(5).

[51] Mousavian M, Chen J, Traylor Z, Greening S. Depression detection from sMRI and rs-fMRI images using machine learning. J Intell Inf Syst 2021;57:395–418.

[52] Trotzek M, Koitka S, Friedrich CM. Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. IEEE Trans Knowl Data Eng 2018;32(3):588–601.

[53] Burdisso SG, Errecalde M, Montes-y Gómez M. A text classification framework for simple and effective early depression detection over social media streams. Expert Syst Appl 2019;133:182–97.

[54] Lin C, Hu P, Su H, Li S, Mei J, Zhou J, Leung H. Sensemood: depression detection on social media. In: Proceedings of ICMR'20 – the 2020 international conference on multimedia retrieval. ACM; 2020, p. 407–11.

[55] Eichstaedt JC, Smith RJ, Merchant RM, Ungar LH, Crutchley P, Preoţiuc-Pietro D, Asch DA, Schwartz HA. Facebook language predicts depression in medical records. Proc Natl Acad Sci 2018;115(44):11203–8.

[56] Bushnell GA, Stürmer T, White A, Pate V, Swanson SA, Azrael D, Miller M. Predicting persistence to antidepressant treatment in administrative claims data: Considering the influence of refill delays and prior persistence on other medications. J Affect Disord 2016;196:138–47.

[57] Bertl M, Shahin M, Ross P, Draheim D. Finding indicator diseases of psychiatric disorders in BigData using clustered association rule mining. In: Proceedings of the 38th ACM/SIGAPP symposium on applied computing. SAC '23, New York, NY, USA: Association for Computing Machinery; 2023, p. 826–33. http://dx.doi.org/10.1145/3555776.3577594.

[58] Purushotham S, Meng C, Che Z, Liu Y. Benchmarking deep learning models on large healthcare datasets. J Biomed Inform 2018;83:112–34.

[59] Medsker LR, Jain L. Recurrent neural networks. Des Appl 2001;5:64–7.

[60] O'Shea K, Nash R. An introduction to convolutional neural networks. 2015, arXiv preprint arXiv:1511.08458.

[61] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Trans Neural Netw 2008;20(1):61–80.

[62] Shang J, Ma T, Xiao C, Sun J. Pre-training of graph augmented transformers for medication recommendation. 2019, URL http://arxiv.org/abs/1906.00346, arXiv:1906.00346 [cs].

[63] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018, arXiv preprint arXiv:1810.04805.

[64] Huang K, Altosaar J, Ranganath R. Clinicalbert: Modeling clinical notes and predicting hospital readmission. 2019, arXiv preprint arXiv:1904.05342.

[65] Ashfaq A, Sant'Anna A, Lingman M, Nowaczyk S. Readmission prediction using deep learning on electronic health records. J Biomed Inform 2019;97:103256.

[66] Hochreiter S, Schmidhuber J, et al. Long short-term memory. Neural Comput 1997;9(8):1735–80.

[67] Chi C-Y, Ao S, Winkler A, Fu K-C, Xu J, Ho Y-L, Huang C-H, Soltani R. Predicting the mortality and readmission of in-hospital cardiac arrest patients with electronic health records: a machine learning approach. J Med Internet Res 2021;23(9):e27798.

[68] Choi E, Xu Z, Li Y, Dusenberry M, Flores G, Xue E, Dai A. Learning the graphical structure of electronic health records with graph convolutional transformer. In: Proceedings of AAAI'20 – the 34th AAAI conference on artificial intelligence. AAAI; 2020, p. 606–13.
[69] Shukla SN, Marlin BM. Interpolation-prediction networks for irregularly sampled time series. 2019, arXiv preprint arXiv:1909.07782.
[70] Wongkoblap A, Vadillo M, Curcin V. Depression detection of Twitter posters using deep learning with anaphora resolution: Algorithm development and validation. JMIR Mental Health 2021.
[71] Mathur P, Sawhney R, Chopra S, Leekha M, Ratn Shah R. Utilizing temporal psycholinguistic cues for suicidal intent estimation. In: Advances in information retrieval – Proceedings of ECIR'2020 : The 42nd European conference on IR research, part ii. Lecture notes in computer science, vol. 12036, Springer; 2020, p. 265–71.
[72] Nadeem A, Naveed M, Islam Satti M, Afzal H, Ahmad T, Kim K-I. Depression detection based on hybrid deep learning SSCL framework using self-attention mechanism: An application to social networking data. Sensors 2022;22(24):9775.
[73] Amanat A, Rizwan M, Javed AR, Abdelhaq M, Alsaqour R, Pandya S, Uddin M. Deep learning for depression detection from textual data. Electronics 2022;11(5):676.
[74] Lam G, Dongyan H, Lin W. Context-aware deep learning for multi-modal depression detection. In: Proceeding of ICASSP'2019 – the 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2019, p. 3946–50.
[75] Chiang H-S, Chen M-Y, Liao L-S. Cognitive depression detection cyber-medical system based on EEG analysis and deep learning approaches. IEEE J Biomed Health Inf 2022.
[76] Lin Y, Liyanage BN, Sun Y, Lu T, Zhu Z, Liao Y, Wang Q, Shi C, Yue W. A deep learning-based model for detecting depression in senior population. Front Psychiatry 2022;13.
[77] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of CVPR'2016 – the 2016 IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[78] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. 2018, arXiv preprint arXiv:1803.02155.
[79] Guo T, Xu Z, Yao X, Chen H, Aberer K, Funaya K. Robust online time series prediction with recurrent neural networks. In: Proceedings of DSAA'2016 – the 2016 IEEE international conference on data science and advanced analytic. IEEE; 2016, p. 816–25.
[80] Jelodar H, Wang Y, Orji R, Huang S. Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. IEEE J Biomed Health Inf 2020;24(10):2733–42.
[81] Ho Y, Wookey S. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. IEEE Access 2019;8:4806–13.
[82] Estonian National Institute for Health Development. RA02: Residents with health insurance and health insurance coverage by sex and county – Tervisestatistika ja terviseuuringute andmebaas. 2020, URL https://statistika.tai.ee/pxweb/en/Andmebaas/Andmebaas_04THressursid_12Ravikindlustatud/RA02.px, Last accessed on 2022-03-04.
[83] Min X, Yu B, Wang F. Predictive modeling of the hospital readmission risk from patients' claims data using machine learning: a case study on COPD. Sci Rep 2019;9(1):2362.
[84] Nolen-Hoeksema S. Gender differences in depression. Curr Dir Psychol Sci 2001;10(5):173–6.
[85] Kessler RC, Birnbaum H, Bromet E, Hwang I, Sampson N, Shahly V. Age differences in major depression: results from the National Comorbidity Survey Replication (NCS-R). Psychol Med 2010;40(2):225–37.
[86] Zhou D, Yuan J, Si J. Health issue identification in social media based on multi-task hierarchical neural networks with topic attention. Artif Intell Med 2021;118:102119.
[87] Nordin N, Zainol Z, Noor MHM, Chan LF. Suicidal behaviour prediction models using machine learning techniques: A systematic review. Artif Intell Med 2022;102395.
[88] Sun J, Han L, Zhao Z. Gene-and evidence-based candidate gene selection for schizophrenia and gene feature analysis. Artif Intell Med 2010;48(2–3):99–106.
[89] Pacheco-Lorenzo MR, Valladares-Rodríguez SM, Anido-Rifón LE, Fernández-Iglesias MJ. Smart conversational agents for the detection of neuropsychiatric disorders: a systematic review. J Biomed Inform 2021;113:103632.
[90] Fielding RT. Representational state transfer (REST), Chapter 5 in (R.T. Fielding): Architectural styles and the design of network-based software architectures (Ph.D.), University of California, Irvine; 2000.
[91] Kroenke K, Spitzer RL. The PHQ-9: a new depression diagnostic and severity measure. Psychiatr Ann 2002;32(9):509–15.
[92] Montgomery SA, Åsberg M. A new depression scale designed to be sensitive to change. Br J Psychiatry 1979;134(4):382–9.

Appendix 4

IV

N. Bignoumba, M. Bertl, S. B. Yahia, and N. Mellouli. Deep magnitude management of clinical code embeddings to predict unplanned hospital readmissions. *PREPRINT (Version 2) available at Research Square*, 2024

Deep Magnitude Management of Clinical Code Embeddings to Predict Unplanned Hospital Readmissions

Nzamba Bignoumba[1*†], Markus Bertl[2,3†], Nedra Mellouli[5],
Sadok Ben Yahia[1,4]

[1*]Department of Software Science, Tallinn University of Technology,
Akadeemia tee 15a, Tallinn, 12618, Estonia.
[2]Department of Health Technologies, Tallinn University of Technology,
Akadeemia Tee 15A, Tallinn, 12618, Estonia.
[3]NextGen Computing Research Group, Unisys, 801 Lakeview Drive Ste
100, Blue Bell, 19422, Pennsylvania, USA.
[4]Centre for Industrial Software, University of Southern Denmark, Alsion
2, Sønderborg, 6400, Denmark.
[5]ESILV, Léonard de Vinci Group, La Défense, Paris, France.

*Corresponding author(s). E-mail(s): nzamba.bignoumba@taltech.ee;
Contributing authors: markus.bertl@taltech.ee;
Nedra.Mellouli@devinci.fr; sadok.ben@taltech.ee;
†These authors contributed equally to this work.

Abstract

The rate of unplanned hospital readmissions is a relevant indicator of the quality of care provided. From a financial point of view, unplanned readmissions are costly for patients and healthcare providers. Awareness of unplanned readmissions helps to mitigate the growth of healthcare costs. Several studies have been carried out to propose models for reducing unplanned hospital readmissions. However, most of these studies do not consider the elapsed time between admissions, when historical medical events are included in said studies. Additionally, the proposed models do not explicitly focus on frequent medical events, such as chronic illnesses, which often lead to unplanned readmissions. Failure to consider the above aspects undoubtedly leads to suboptimal prediction of unplanned readmissions. To remediate, we introduce a deep sequential learning model called '*Deep Magnitude Management*' (D2M) that handles sequences of admissions according to their corresponding date and incorporates a mechanism that allows it to focus explicitly on frequent medical events. To provide effective evidence, we compare the performance of D2M with the state-of-the-art models and conduct various ablation studies using the MIMIC-3 database. Furthermore, we provide a series of graphs for explainability purposes.

Keywords: Artificial Intelligence, Deep Learning, Decision Explainability, Healthcare, Hospital Readmission Prediction, Support System.

1 Introduction

Hospital readmissions represent a major challenge for health systems worldwide, both in terms of patient outcomes and efficiency in the allocation of health resources. The rehospitalization of patients is not only financially burdensome but can also result in increased morbidity and mortality rates, making it a critical issue in healthcare management. Readmissions are detrimental for several compelling reasons. First, they place additional physical and emotional burdens on patients who have already experienced the stress and discomfort of an initial hospital stay. Returning to the hospital often signifies deterioration in health and can lead to further suffering and anxiety [1]. Second, frequent readmissions can significantly increase healthcare costs, straining healthcare systems and resources. This financial burden affects not only hospitals and insurers, but also the patients themselves, who may have to pay out of their own pockets. A study conducted in [2] showed that unplanned readmissions caused higher costs, ranging from $\$13,424$ to $\$21,448$ per patient. Moreover, unplanned readmissions suggest potential issues with the quality of care or discharge planning. Last, unplanned hospital readmissions can result in a cycle of care fragmentation, where patients receive disjointed or inadequate treatment, leading to a prolonged and compromised recovery process [3]. Thus, reducing unplanned readmissions is not only essential for improving patient well-being and treatment outcomes but also for optimizing the effectiveness and sustainability of healthcare delivery and enabling evidence-based medicine and effective hospital management [4]. As a response to this challenge, the application of deep learning techniques has emerged as a promising avenue for predicting and mitigating unplanned hospital readmissions.

In the literature, unexpected readmission occurring within 30 days of hospital discharge is considered unplanned hospital readmission or simply hospital readmission [5]. The terms “hospital readmission” and “unplanned hospital readmission” are used interchangeably. This paper explores the potential of deep learning to predict, at hospital discharge, whether a patient may be subject to unplanned readmission within 30 days. We delve into the complexities of this problem, considering the multifaceted factors that influence a patient’s likelihood of returning to the hospital after discharge. By harnessing the power of deep learning, which is especially suitable for complex tasks such as this [6], and leveraging patients’ Electronic Medical Records (EHRs), we aim to enhance our predictive model and ultimately reduce the burden of avoidable readmissions on healthcare systems.

There are a number of digital decision support systems that can do in-depth analyses quickly and with digital data from patients' electronic health records (EHRs) [7, 8, 9]. These models are repurposed versions of state-of-the-art models used in the medical context, such as natural language processing (NLP), computer vision, and time series forecasting, to name a few. As unplanned readmission prediction may be conditioned by heterogeneous data, different types of models are usually combined and trained in an end-to-end fashion [10]. For the proposals relying partially or fully on a sequence of categorical features such as prescription, medication, and procedure, we note that Transformers and Recurrent Neural Networks (RNNs) are the most preferred [11, 12, 13, 10]. This is justified by the fact that they are proven to be more efficient than other models on sequence modeling tasks, such as emotion detection [14] or machine translation [15].

Although several functional RNN and Transformer-based models [10, 11, 16] have been developed to address the unplanned readmission prediction challenge, we have noticed that most of them do not take into consideration one of the relevant factors, namely the time elapsed between admissions when dealing with historical medical events. This can be problematic because the model will assign the same level of importance to historical and current medical events. Furthermore, these models do not incorporate an explicit mechanism that focuses on frequent clinical codes (diagnoses, procedures, and medications) across admissions or visits. Given that frequent medical events, such as chronic diseases, often result in patients returning to the hospital [17], we assume that an explicit mechanism focusing on frequent clinical codes in general and diseases (diagnoses) in particular should be built into the model's core to enable accurate prediction. To address these gaps, we then proposed a deep sequential neural network-based model called *Deep Magnitude Management* (D2M), which processes the content of sequential admissions based on their respective dates. Additionally, D2M incorporates an explicit information transfer mechanism that allows it to focus on frequent clinical codes while capturing their evolution over time. Note that our model also has the advantage of working with all types of patients rather than being dedicated to a specific patient cohort. Indeed, while remaining accurate, D2M may be an alternative for hospitals that do not have sufficient funds and/or resources to develop a model for each cohort of patients. Below, we outline the article's main contributions:

- We introduce a deep sequential learning model called *Deep Magnitude Management* (D2M) that overcomes the limitations of previous studies, namely the inability to account for elapsed days between admissions when processing historical medical events and the lack of a mechanism explicitly focused on frequent medical events, such as chronic diseases, which are common causes of readmission;
- We show through extensive experiments and comparisons with the state-of-the-art models that by overcoming the identified limitations, D2M makes it possible to increase the accuracy of hospital readmission prediction;
- We show how the reasoning of D2M can be explained;
- And lastly, we provide a first step towards tackling the challenge of hospital readmissions in healthcare.

We use Medical Information Mart for Intensive Care (MIMIC-3) [18] to carry out the experiments. The results obtained show that D2M is more effective in predicting hospital readmission than competing models. We believe that D2M will certainly help healthcare professionals, decision-makers, and ultimately patients by improving their outcomes.

We organize the remaining portions of this paper as follows: We present the background work in Section 2. Section 3 provides a formal description of our model. In Section 4, we examine the model's performance and compare it with the state-of-the-art models. We also conduct various ablation studies to highlight the significance of D2M strategies and provide graphs to clarify the predictions. In Section 5, we discuss the advantages and disadvantages of D2M. Finally, Section 6 recounts the takeaways and contributions of this paper and sketches pathways for future work.

2 Background

The adoption of EHR by more and more healthcare institutions enables the use of AI models for medical problem-solving. The combination of computer science and medical knowledge has made it possible to develop functional models for tasks such as breast cancer prediction [19], mortality prediction [20], and readmission prediction [21], which is the task of interest in this study. Indeed, several approaches have been proposed to solve the unplanned readmission problem. One of the most common is to tackle it as an NLP task. The varied and rich content of the EHR makes this possible. Indeed, EHR may include structured data like categorical data (clinical codes), time series data, and patient demographics, and unstructured data like clinical notes, or medical images.

Unstructured clinical notes alone may contain various relevant information about the patient's condition, treatment history, and visit information. For this reason, several studies have used them exclusively to predict unplanned readmissions. For instance, in [22] they used clinical notes for predicting heart failure readmission. Instead of building a model to predict readmissions for a cohort of patients, the authors in [13] and [23] proposed generic models that predict readmissions based on clinical notes. Although authors in [13] used a transformer encoder for better data representation, they did not take the time factor into account. Additionally, the data quality of unstructured text data like clinical notes is often insufficient for AI training [24].

As readmission prediction is generally confronted with unbalanced class problems, various works introduced different loss function strategies [11, 12] to overcome this problem. Unlike the aforementioned models, they used heterogeneous data such as laboratory events, diagnoses, procedures, and patient demographics. In [25], authors used the synthetic minority oversampling technique (SMOTE) [26] to balance the classes during learning. As the deep learning model they used may discard some relevant features during the forward pass, they explicitly injected other engineering features to reinforce the learning. A limitation of this work is that the medical history is not considered. In [27], they have shown that with an effective combination of knowledge (engineering) and data-driven features, machine learning can overtake deep learning methods in predicting unplanned readmissions. In contrast to [25], they included the patient's data history while weighting it by the elapsed time. Although this method addresses the readmission problem as well as we do, it is only suitable for patients with chronic obstructive pulmonary disease. Another work similar to ours, i.e. a work that takes into account the time elapsed between visits and incorporates attention mechanisms, was proposed in [28]. However, the proposed method, Timeline, was designed for breast cancer patients and does not include a mechanism that explicitly focuses on frequent medical events. To predict the readmission rate of heart failure patients, in [10] they implemented a content model using a vanilla RNN variant, to model the sequential aspect of visits. The problem with vanilla RNNs is that they treat visits as events that occur at regular time intervals. This is an incorrect assumption, as the time between visits may differ. We note that, unlike all the models we have mentioned so far, the content model has the advantage of explicitly leveraging patient similarities. The authors in [29] built a model that exclusively predicts the unplanned readmission of patients who have been treated in the emergency department. In [30], the authors proposed to predict 30-day unplanned ICU readmission in heart failure patients by taking into account historical medical events and their associated timestamps. To integrate temporal information (timestamps), they mapped data extracted from the EHR to event logs. Despite their use of historical medical events and associated timestamps, their method does not apply to patients with a single admission. This may be limited in real-world scenarios where we would like to predict the readmission of a patient who has been admitted for the first time and therefore has no historical medical events. Our model has the advantage of being able to make predictions even for patients without historical medical data. The authors of [31] created knowledge graph embeddings from biomedical ontologies to make it easier to get meaning from clinical features and store how they relate to each other. Although this approach makes it possible to improve the accuracy of ICU readmission risk at different stages (before, during, after), in particular during, the model is based solely on data from the current admission. Similar work based on knowledge graphs has also been carried out in [32] for ICU readmission prediction.

Since the patient's daily behaviour and lifestyle after discharge from the hospital may be a factor in readmission, the authors in [33, 34] used data collected from patients' wearable devices to predict possible readmission at each stage of patient treatment. In [35], they adopted an alternative approach that continuously monitored and predicted readmission risk on a daily basis. Other works, such as [36, 37], have also used data collected from wearable devices to predict the risk of readmission.

Although the works identified have obtained functional results, we have noted that most of them do not take into account the time elapsed between admissions during the learning process. For those that do take this aspect into account, they do not incorporate an explicit mechanism focusing on frequent medical events or only predict readmission for a specific cohort of patients.

3 Method

Partially or fully conditioned by textual or alphanumeric data, the prediction of unplanned readmissions is generally approached as a natural language processing

(NLP) task [10, 13], in particular as a sequential modelling task. Therefore, several models, such as RNN and Transformer, have been redesigned to address the readmission prediction challenge [11, 13, 28]. Although functional, the majority of these models do not take into account the time elapsed between admissions during their decision-making process. For those that do, they do not integrate a mechanism that explicitly focuses on frequent medical events that may be the cause of readmission. Aware of these limitations, the D2M proposed model aims to fill these gaps. D2M is a sequential model that processes the sequence of medical admissions based on their date while focusing on frequent medical events (clinical codes) through an explicit information transfer mechanism. We believe that our model will better equip healthcare professionals to anticipate possible unplanned readmissions, therefore enabling them to improve healthcare delivery.

In what follows, we first introduce data notation, then we present the different components of deep magnitude management (D2M). Finally, we present the readmission prediction classifier.

3.1 Data notation

Let, $\mathcal{D} = \{\mathbf{v}_n^1 \cup \mathbf{v}_n^2 \cup \cdots, \mathbf{v}_n^{(J-1)} \cup \mathbf{v}_n^J; \mathbf{s}_n; \mathbf{d}_n\}_{n=1,2,\cdots,N/1\leq j\leq J}$ where N is the number of samples and J the number of admissions. $\mathbf{v}_n^j = \{c2_1^j, c1_2^j, \cdots, cl_k^j\}_{k=1,\cdots,K}$ is the unordered set of clinical codes (procedures, prescriptions, and diagnoses) of the n-th sample and the j-th admission. K is the number of clinical codes per admission. cl_k^j is a clinical code associated with the integer l at the index position k. $1 \leq l \leq L$ where L is the number of distinct clinical codes (also called vocabulary size). We call the union of all admissions $\mathbf{v}_n^1 \cup \mathbf{v}_n^2 \cup \cdots, \mathbf{v}_n^{(J-1)}$ the historical medical events, and $\mathbf{v}_n^J$ the current medical event. $\mathbf{s}_n = [s^1, \cdots, s^{J-2}, s^{J-1}]$ is the elapsed time vector. s^j represents the time elapsed between the j-th and $(j+1)$-th admission. $\mathbf{d}_n = [d_1, d_2, d_3, d_4, d_5]$ is the vector of demographic codes (sex[1], age) and additional information codes of the J-th admission (the admission type, the length of stay and the insurance type) for the *n-th* sample. We call $\mathbf{d}_n$ complementary features.

3.2 Deep magnitude management (D2M)

Unlike existing sequential models such as RNN and Transformer, in D2M, the condition allowing information transfer from one sequence (admission in our case) to another is defined in advance and integrated into the model's computational scheme. When the condition is met, the transfer of information, controlled by two main aspects, namely the time elapsed between admissions and their similarity, can take place. The condition that D2M must meet to transfer the information is as follows: **information must flow from one admission to another if they have at least one common clinical code**. This strategy allows D2M to focus more on frequent medical events such as chronic diseases that may cause readmission. The D2M architecture shown in Figure 1 encompasses an order management component, an embedding layer, a magnitude management layer, a feedforward neural network and a classifier. Each of

---

[1] Here, sex refers to a set of biological attributes that are associated with physical and physiological features (e.g. chromosomal genotype, hormonal levels, internal and external anatomy).

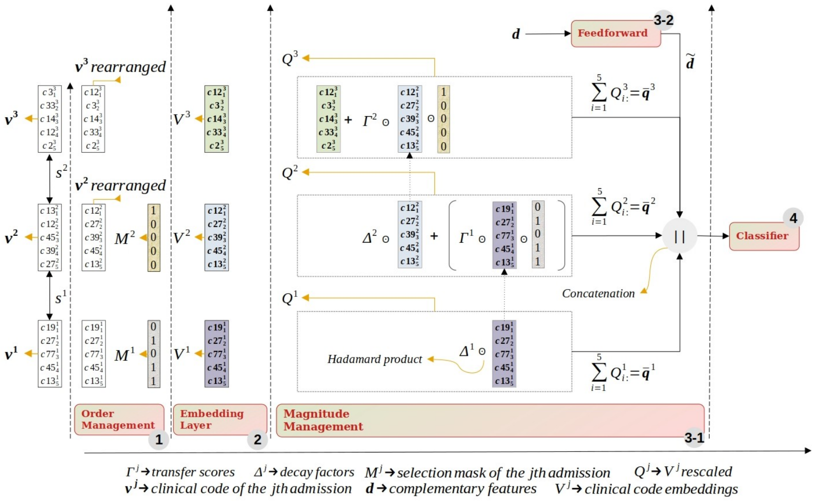


Fig. 1: D2M architecture: representation of the forward pass process with a patient having 3 admissions. No information transfer is applied to calculate Q^1 because V^1 is the representation of the first admission. We did not apply decay factors to V^3 since its clinical code embeddings are those of the current admission. The middle block in the magnitude management illustrates how the intermediate admissions must be processed if we have more than 2 admissions.

them is described in the following sections. The subscript n is omitted in the following sections to simplify the formulas (except in Algorithm 1).

3.2.1 Order management algorithm

The order management algorithm provides the items needed to implement the condition related to the information transfer. It rearranges the set of clinical codes $\mathbf{v}^j$ $(j \neq 1)$ and calculates a selection mask $M^j \in \{0,1\}^{K \times 1}$ $(j = 1, ..., J-1)$ such as:

$$m_k^j = \begin{cases} 1 \text{ if } cl_k^j = cl_k^{j+1} \\ 0 \ otherwise \end{cases} \tag{1}$$

where, $cl_k^j \in \mathbf{v}^j$ if $j = 1$, otherwise $cl_k^j \in \mathbf{v}^j \ rearranged$. $cl_k^{j+1} \in \mathbf{v}^{j+1} \ rearranged$ (see the order management block in Figure 1 for illustration). The selection mask M^j will be responsible for selecting the clinical codes in $\mathbf{v}^j$ that should be weighted and added to $\mathbf{v}^{j+1}$. Algorithm 1 lists the different order management instructions. Its complexity is: $\mathcal{O}(N \times (J-1) \times 2K)$ where N is the number of samples, J the number of admissions per sample, and K the number of clinical codes per admission. **It should be noted**

that order management (and hence the proposed model) is only applicable if there is no sequential order in the input sequence.

Algorithm 1 Order management

```
Require: $\{\mathbf{v}_n^1 \cup \mathbf{v}_n^2 \cup \cdots, \mathbf{v}_n^{(J-1)} \cup \mathbf{v}_n^J\}_{n=1,2,\cdots,N}, \mathbf{M} \in \{0\}^{N\times J-1\times K\times 1}, N, J, K$  ▷ $\mathbf{M}$ is the initial selection mask tensor.
for $n = 1$ to $N$ do  ▷ $N$ is the number of samples.
    for $j = 1$ to $(J-1)$ do  ▷ $J$ is the number of admissions per sample.
        $\mathbf{t}$, $\mathbf{m}$=[],[]  ▷ Intermediate variables.
        for $cl_k^j$ in $\mathbf{v}_n^j$ do
            if $(cl_k^j \in \mathbf{v}_n^{j+1})$ and $(cl_k^j \neq 0)$ then  ▷ Zero is used as a filler value, so we skip the calculation when it is encountered.
                $\mathbf{t}.add(cl_k^j)$  ▷ add $cl_k^j$ in $\mathbf{t}$
                $\mathbf{m}.add(1)$
                $\mathbf{v}_n^{j+1}.remove(cl_k^j)$  ▷ remove $cl_k^j$ in $\mathbf{v}_n^{j+1}$.
            else
                $\mathbf{t}.add(0)$
                $\mathbf{m}.add(0)$
            end if
        end for
        if $\sum_{i=1}^{len(\mathbf{t})} t_i > 0$ then  ▷ If at least one element of $\mathbf{t}$ is non-zero.
            for $t_i$ in $\mathbf{t}$ do
                if $(t_i == 0)$ and $(len(\mathbf{v}_n^{j+1}) > 0)$ then
                    $cl_*^{j+1} \leftarrow \mathbf{v}_n^{j+1}.randomChoice()$  ▷ Randomly select an element $cl_{*=1,\cdots,K}^{j+1} \in \mathbf{v}_n^{j+1}$.
                    $t_i \leftarrow cl_*^{j+1}$
                    $\mathbf{v}_n^{j+1}.remove(cl_*^{j+1})$
                end if
            end for
            $\mathbf{v}_n^{j+1} \leftarrow \mathbf{t}.padd(0, K)$  ▷ Zero padding to obtain $\mathbf{t} \in \mathbb{R}^K$. $\mathbf{v}_n^{j+1}$ is rearranged.
            $\mathbf{m} \leftarrow \mathbf{m}.padd(0, K)$  ▷ Zero padding to obtain $\mathbf{m} \in \{0,1\}^K$.
            $\mathbf{M}[n, j, :, 1] \leftarrow \mathbf{m}^\intercal$  ▷ Save the corresponding selection mask.
        end if
    end for
end for
return $\mathbf{M}$
```

Once the selection mask has been calculated, the next step is to map each clinical code to its embedding version. In the remainder of the article, we consider only two admissions $\mathbf{v}^j$ and $\mathbf{v}^{j+1}$. $\mathbf{v}^j$ refers to the past admission (historical medical event) and $\mathbf{v}^{j+1}$ to the current admission. In the case of more than two admissions, Figure 1 illustrates how they are processed.

3.2.2 Clinical code embeddings

To encode each clinical code into a vector that carries its semantics and underlying relationships with other codes, we passed them through an embedding layer:

$$V^j, V^{j+1} = \rho_\theta(\mathbf{v}^j \parallel \mathbf{v}^{j+1}_{rearranged}) \tag{2}$$

where $\rho(.)$ is the embedding layer. $(. \parallel .)$ is the concatenate symbol and θ the learnable embedding parameters. V^j and $V^{j+1} \in \mathbb{R}^{K\times \mathsf{d}}$ are matrices where each row $V^j_{k:}$ represents the embedding version χl^j_k of the clinical code cl^j_k. d is the dimension of the embedding space. It should be noted that a single embedding layer is used because we want a better representation of clinical codes. If we consider past and present medical events as the overall context, the embedding process will work better because more clinical codes will occur together, leading to more accurate clinical code embeddings.

In the following section, we describe how these clinical code embeddings are passed through the magnitude management layer for sequential modelling of admissions.

3.2.3 Magnitude management

Magnitude management is a sequential layer that transfers information contained in admission $\mathbf{v}^j$ to admission $\mathbf{v}^{j+1}$. The clinical code in which the information is extracted in $\mathbf{v}^j$ must also be part of the admission $\mathbf{v}^{j+1}$. This strategy allows the model to better capture frequent clinical codes in general and chronic diseases in particular.

We call this layer 'magnitude management' because the explicit transfer of information leads to a change in the magnitude of the clinical code embeddings. The information transfer process is controlled by the similarity between successive admissions (modelled by similarity scores) and the time elapsed between these admissions (modelled by decay factors).

Similarity scores

Similarity scores are values that quantify the level of similarity between two successive admissions. Indeed, a patient may be diagnosed with the same disease on two successive admissions, but the medical context of these admissions may be different. It is therefore important, using similarity scores, to compare the medical context of these two successive admissions before transferring any information. We distinguish two types of similarity scores: the admission similarity score and the mask similarity score.

The admission similarity score measures the similarity between the content of successive admissions. In other words, it measures the degree of similarity between the patient's state of health at admission j and $j+1$. The admission similarity score is obtained as follows:

$$\phi^j = \frac{\bar{\mathbf{v}}^j (\bar{\mathbf{v}}^{j+1})^\mathsf{T}}{||\bar{\mathbf{v}}^j||.||\bar{\mathbf{v}}^{j+1}|| + \epsilon} \tag{3}$$

$$\bar{\mathbf{v}}^* = \sum_{n=1}^{K} V^*_{n:}; * \in \{j, j+1\} \tag{4}$$

where $\phi^j \in [-1, 1]$ is the admission similarity score between the j-th and $(j+1)$-th admission. $||.||$ is the Euclidean norm symbol. $\bar{\mathbf{v}}^* \in \mathbb{R}^{\mathsf{d}}$ called admission embedding is the sum of clinical embedding $V^*_{k:} = \chi l^*_k$. As some patients may not have any historical medical events, i.e. $||\bar{\mathbf{v}}^j|| = 0$, we add ϵ to the denominator in Equation 3 to prevent division by zero.

The mask similarity score aims to encode the similarity of two successive admissions based on their selection mask. Unlike the admission similarity score, which encodes similarity between admissions at the content level, the mask similarity score encodes admission similarity at the structure level. This can be viewed as comparing two people on the basis of their physical appearance (mask similarity score) and their morality (admission similarity score). In terms of matrix representation, let us imagine that we have a selection mask whose values are all equal to one, which means that all clinical codes appearing at admission j appear at admission $j+1$. The patient's condition at these two admissions has therefore probably remained the same. The mask similarity score is obtained through the non-linear and non-parametric transformation:

$$\omega^j = \begin{cases} \sigma(\sum_{n=1}^{K} M_n^j) & if \;\; \sum_{n=1}^{K} M_n^j \neq 0 \\ 0 & otherwise \end{cases} \tag{5}$$

where σ is the *sigmoid* activation function. Equation 5 maintains the mask similarity score scale between $[0, 1]$.

Decay factors

We assume that a medical event that occurred a long time ago should not be interpreted in the same way as a medical event that occurred in the present. Therefore, we calculated a decay factor matrix whose values model the importance of each clinical code over time. Each decay factor value is specific to a clinical code, as clinical codes such as diagnoses vary differently over time. For example, COVID-19 may take longer to heal than the flu. Formally, if an admission has K clinical code embeddings, we first concatenate them individually with the elapsed time value (see Equation 7). The result of this concatenation is linearly transformed and passed through an exponential decay function (Equation 6) to obtain the corresponding decay factor matrix:

$$\Delta^j = exp\{-max(0, \delta^j W_\Delta + B_\Delta)\} \tag{6}$$

$$\delta^j = [s^j \parallel V^j_{1:}, \cdots, s^j \parallel V^j_{K:}]^\intercal \tag{7}$$

$V^j_{k:}$ which is the k-th row of V^j represents the clinical code embedding χl^j_k. s^j is the elapsed time between the j-th and $(j+1)$-th admission. $\Delta^j \in \mathbb{R}^{K \times 1}$ is the decay factor matrix. Each value Δ^j_k is a decay factor value associated with the clinical embedding χl^j_k. $W_\Delta \in \mathbb{R}^{+(\mathsf{d}+1)\times 1}$ and $B_\Delta \in \mathbb{R}^{K \times 1}$ are learnable parameters.

The values of the decay factor matrix $\Delta^j \in \,]0, 1]$ because $\lim_{* \to +\infty} exp(-max(0, *)) = 0$ and $exp(0) = 1$. Verbally, the higher the value of the elapsed time s^j between two consecutive admissions, the closer we expect the values of the decay factor matrix Δ^j to be to zero [2]. Therefore, if a set of clinical codes was

---

[2] We say expect because decay factors also depend on clinical code embeddings.

recorded long ago, their vector form (clinical code embedding) will be multiplied by decay factors whose values are close to zero. This strategy allows historical clinical codes to be used as additional information without significantly impacting prediction if recorded long ago. By analogy, the physician can consult the medical history in the patient's electronic medical record before making a final decision. Based on the patient's current examinations, the physician will use the relevant medical history as additional information to understand better what the patient is suffering from. Although these are not the only factors, the relevance of a historical medical event depends on when it was recorded, how often it occurred (e.g. was it a chronic disease?), and whether it is a part of current medical events.

In the following paragraph, we show how the decay factor matrix is combined with the similarity scores to control the information transfer process.

Information transfer process

The decay factor matrix, is combined with the similarity scores to guide the transfer of information from one admission to the next. For the patient's first admission, i.e. the historical medical event, as we cannot transfer any information (because there was no previous admission), we simply weighted the clinical code embeddings:

$$Q^j = \Delta^j \odot V^j \tag{8}$$

where $\odot$ is the Hadamard product and $Q^j \in \mathbb{R}^{K \times \mathsf{d}}$ represents V^j rescaled (also called latent representation of $\mathbf{v}^j$). The underlying intuition here is to reduce the magnitude of clinical code embeddings $\chi l_k^j \in V^j$ if they took place a long time ago, and to keep their magnitude almost unchanged if they took place in the recent past.

For the second admission, which is the current medical event (since we are only considering two admissions, i.e. $J = 2$), its rescaled representation is obtained as follows:

$$Q^{j+1} = V^{j+1} + \Gamma^j \odot M^j \odot V^j \tag{9}$$

$$\Gamma^j = tanh([\Delta^j \parallel \phi^j \parallel \omega^j]W_\Gamma + b_\Gamma) \tag{10}$$

where $\Gamma^j \in [-1, 1]^{K \times 1}$, the information transfer score matrix, is a non-linear transformation of the concatenation of the decay factor values Δ^j and similarity scores ϕ^j and ω^j. $tanh(.)$ is the tangent hyperbolic function. $W_\Gamma \in \mathbb{R}$ and $b_\Gamma \in \mathbb{R}$ are learnable parameters.

In the second term of Equation 9, M^j selects the clinical code embeddings in V^j that are in V^{j+1}, and Γ^j weights these clinical code embeddings. The operand + carries out the information transfer from V^j to V^{j+1}. Each row $Q_{k:}^{j+1}$ of Q^{j+1} is either an initial clinical code embedding χl_k^{j+1} or a new clinical code embedding $\chi l_k^{j+1} + \chi l_k^j \times \Gamma_k^j$ (see the magnitude management block in Figure 1 for illustration).

Each transfer score Γ_k^j controls the quantity and quality of information that should be transferred from χl_k^j to χl_k^{j+1}. Quality here refers to the sign of Γ_k^j. When this latter is negative, it tends to decrease the magnitude of χl_k^{j+1} via the operation $\chi l_k^{j+1} + \chi l_k^j \times \Gamma_k^j$. Otherwise, it tends to increase [3] the magnitude of χl_k^{j+1}. The underlying

---

[3] We say "tends to decrease (or increase)" because the signs of $\chi l_k^j \times \Gamma_k^j$ also depend on χl_k^j.

intuition is that we want to increase the magnitude of a diagnosis embedding when the corresponding disease worsens and decrease it when the disease improves. This approach allows the model to capture disease progression over time, providing a more accurate representation of the patient's health status.

Once the information transfer process is complete, the latent representation of the j-th and $(j+1)$-th admission, Q^j and Q^{j+1} respectively, will be combined as follows:

$$\mathbf{q} = (\bar{\mathbf{q}}^j \parallel \bar{\mathbf{q}}^{j+1}) \tag{11}$$

$$\bar{\mathbf{q}}^* = \sum_{n=1}^{K} Q^*_{k:};\ * \in \{j, j+1\} \tag{12}$$

where $\bar{\mathbf{q}}^* \in \mathbb{R}^{1\times \mathsf{d}}$ is the sum of all rows in Q^*. $\mathbf{q} \in \mathbb{R}^{1\times 2\mathsf{d}}$ is the latent patient's condition vector. It will be combined with patient demographics and additional admission information and then fed into the classifier.

3.3 Classifier

To find out whether admission will result in an unplanned readmission, we combine the latent patient's condition vector $\mathbf{q}$ with the latent complementary features $\tilde{\mathbf{d}}$:

$$\tilde{\mathbf{d}} = f_\alpha(\mathbf{d}) \tag{13}$$

where f is a feedforward neural network with one layer and α its set of learnable parameters. Therefore, the likelihood that a patient will be readmitted is:

$$\hat{y} = g^1_{\beta_1} \circ g^2_{\beta_2} \cdots \circ g^S_{\beta_S}((\mathbf{q} \parallel \tilde{\mathbf{d}})) \tag{14}$$

where the concatenation $(\mathbf{q} \parallel \tilde{\mathbf{d}})$ is the latent representation of the patient. $\circ$ is the composition symbol. g^2,$g^3 \cdots$, $g^{(S)}$ are $(S-1)$ stacked dense layers with *relu* as activation function. g^1 is the last feedforward neural network with *sigmoid* as activation function. β_1,β_2, $\cdots$,β_S are the learnable parameters. $\hat{y} \in [0,1]$ is the likelihood that a patient will be readmitted.

As a loss function, we used the binary cross entropy, defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}[y_n * \ln(\hat{y}_n) + (1-y_n) * \ln(1-\hat{y}_n)] \tag{15}$$

where y is the true value. It should be noted that other loss functions, such as weighted binary cross entropy and focal loss [38], which are designed for unbalanced classification problems, were tested, but we did not obtain any improvement.

4 Experimental evaluation

This section is devoted to the empirical evaluation of the proposed model. First, we present the dataset and the data extraction process. Then, we benchmark the model against state-of-the-art models and carry out various ablation studies to demonstrate the effectiveness of strategies implemented in D2M. Finally, some graphs are provided to explain how the model makes decisions.

We coded the proposed model using the Python 3.0 programming language and the machine learning libraries Keras 2.9.0 and TensorFlow 2.9.2. All remaining pre-processing and performance evaluations were done with NumPy, Pandas, and Scikit-Learn libraries. Finally, we ran the code on a computer cluster with the following characteristics: The AMD Threadripper 3960X is a 24-core, 48-thread processor with 128 GB of memory. It is paired with an NVidia 3090 GPU with 24 GB of graphics memory.

4.1 Dataset

To conduct the study, we used MIMIC-3, a large and freely available database comprising disidentified health-related data from forty thousand patients who were admitted between 2001 and 2012 to the critical care units of the Beth Israel Deaconess Medical Center. The data was collected from:

- Archives from critical care information systems (namely MetaVvision and CareVue)
- Hospital EHR databases
- Social Security Administration Death Master File

Indeed, MIMIC is one of the most widely used databases in the literature for benchmarking machine learning models designed for medical problems. We used hospital EHR data to obtain billing-related information such as patient demographics, in-hospital mortality and clinical codes in the format of the International Classification of Disease, 9th Edition (ICD-9). From the admissions table, we initially extracted data from $46,520$ patients. We removed patients under 18 years old. In hospitals, death excludes readmission, so admissions with death are eliminated. Out-patients were excluded. In-patients for whom we did not have data on diagnosis, prescriptions, and procedures were also excluded. Finally, patients admitted only once and whose date of death (outside the hospital) is unknown were removed. It is worth pointing out that the deceased patients filtered out are those who die in the hospital and not outside the hospital. After applying these filters, we obtained $15,344$ samples, of which $2,600$ (10.20%) are linked to unplanned readmissions (positive cases) and $12,744$ are not (negative cases). The following cases are considered unplanned readmission: patients readmitted within 30 days after discharge from the hospital; and patients who died within 30 days of discharge from the hospital. It should be noted that a hospital readmission unrelated to diagnoses made on the previous admission is not classified as unplanned hospital readmission. The maximum number of admissions per patient is 2, which is approximately the average number of admissions per patient across the entire database. The data extraction process is illustrated in Figure 2 and the data description is summarized in Table 1.

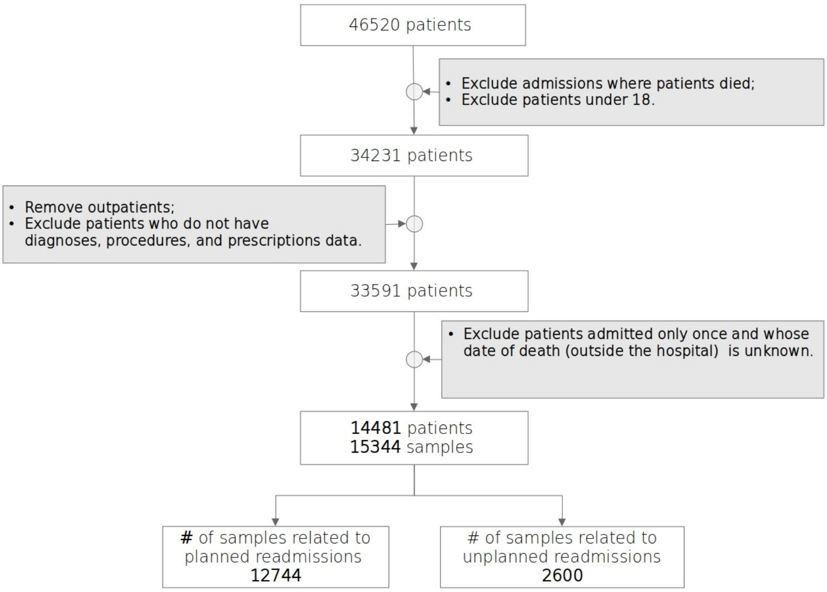

46520 patients
Exclude admissions where patients died;
Exclude patients under 18.
34231 patients
Remove outpatients;
Exclude patients who do not have diagnoses, procedures, and prescriptions data.
33591 patients
Exclude patients admitted only once and whose date of death (outside the hospital) is unknown.
14481 patients
15344 samples
# of samples related to planned readmissions
12744
# of samples related to unplanned readmissions
2600

Fig. 2: Data extraction process.

Table 1: Data description

Features	Values
# of clinical codes	8,422
# of clinical codes per admission	107
Average # of admissions per patient	2
Average age	68
# of admission types	3
# of insurance types	5
Average length of stay	11 days

4.2 Model setting

We performed an extensive grid search to obtain the optimal hyperparameters of the model. The retained values of these hyperparameters are as follows: the embedding dimension is 50 (see Equation 2); a dropout of 0.8 is applied on $\bar{\mathbf{q}}^{j+1}$; the dense layer dedicated to extracting information from complementary features has 15 units and its activation function is *relu* (see Equation 13); and finally, the classifier is composed of two stacked dense layers with 45 units and 1 unit respectively (see Equation 14).

We used the Adam optimizer to fit the model's parameters during training. The learning value rate was set to 0.001. The number of epochs and the batch size value were set to 100 and 200, respectively. We used 5-fold cross-validation to train and evaluate our model and competitors. Technically, during five loops, 4/5 of the dataset was used for training (validation included 10%) and the remaining 1/5 for testing. The hyperparameters and training parameters of the competing models are presented in Appendix A.

4.3 Readmission prediction performances

We usher in this section with a focus on in-hospital readmission prediction accuracy. Since the dataset is unbalanced, the Area Under the ROC Curve (AUC) and the Area Under the Precision-Recall Curve (AUPRC) were used as metrics. The proposed model and the competitors have been run over 5-fold cross-validation. The average AUC and AUPRC are reported in Table 2. The confusion matrices from a 1-fold cross-validation are also provided in Appendix B.

Long Short-Term Memory (LSTM) [39], GRU-Decay [4] [20] (GRU stands for Gated Recurrent Unit), Retain [40], Timeline [28], Transformer [41] (only the encoder is used), and Logistic Regression (LR) are the competing models against which we evaluated the proposed model for benchmarking purposes. Except for the LR model, whose admissions are encoded in a one-hot encoding format, the admissions for all other models are the sum (or weighted sum) of clinical code embeddings.

Table 2: AUC and AUPRC scores over 5-fold cross-validation of competing models vs ours.

Models	AUC	AUPRC	# of parameters
GRU-Decay [20]	0.691 ± 0.010	0.364 ± 0.007	$921,865$
LR	0.641 ± 0.008	0.303 ± 0.005	$\mathbf{119,40}$
LSTM [39]	0.673 ± 0.009	0.343 ± 0.008	$1,018,356$
Retain [40]	0.686 ± 0.012	0.348 ± 0.010	$723,060$
Timeline [28]	0.692 ± 0.010	0.361 ± 0.011	$749,555$
Transformer [41]	0.684 ± 0.015	0.358 ± 0.014	$637,756$
D2M	$\mathbf{0.705 \pm 0.007}$	$\mathbf{0.372 \pm 0.012}$	$\mathbf{601,787}$

± Standard deviation.

From Table 2, we observe that D2M achieves the best AUC and AUPRC scores. This confirms the effectiveness of the different strategies we have implemented. We find that all models incorporating a mechanism for modelling the time elapsed between admissions, including Timeline and GRU-decay, obtain the best results after our model. This shows just how crucial it is to consider the time factor.

Our model's emphasis on frequent clinical codes can be seen as an attention mechanism. Consequently, if we compare this attention with that of Retain (attention at

---

[4] Unlike the original version proposed in [20], it does not incorporate the imputation strategy.

admission level) and Transformer (attention at clinical code level), we can assume that ours, in the case of readmission prediction, is more effective because of the higher scores we obtained. The fact that the elapsed time factor is a part of our attention mechanism may explain this superiority. Indeed, our model focuses on frequent clinical codes while considering their evolution over time. This is not the case with Retain and Transformer. Although Timeline incorporates an attention mechanism that also exploits the elapsed time factor, it does not explicitly focus on frequent clinical codes. This may explain its underperformance compared to D2M.

LSTM and Transformer are less accurate than GRU-decay, Timeline and D2M because they treat medical events as if they always occurred at regular time intervals. This is the wrong approach because the time between admissions is decidedly irregular. For instance, in Transformer, the position-encoding vectors that encode the order of admissions assume a regular interval of one day between admissions.

It is also observed that models such as LSTM and LR, which do not incorporate attention mechanisms, perform less well than the other models including Retain, Timeline, Transformer and D2M, which all incorporate attention mechanisms. This underperformance highlights the crucial role played by attention mechanisms when processing complex and heterogeneous data such as medical data.

Unsurprisingly, the LR model, the unique non-deep learning model, is the least accurate. Indeed, while the LR model requires, in some cases, additional feature engineering to be more efficient, deep learning models can extract hidden features and therefore do not need additional feature engineering. This capability makes them, in most cases, more efficient than traditional machine learning and statistical models.

Regarding the number of parameters, Table 2 shows that D2M is the deep learning model with the fewest parameters. With climate change a major concern and information technology (IT) playing an important role in achieving the United Nations' sustainable development goals [42], it is important to offer functional and green models [43]. This green aspect involves providing models with a relatively small number of parameters while remaining sufficiently accurate. In conclusion, if we have to make a compromise between performance and ecology, our model seems to be the best choice for the hospital readmission prediction task.

4.4 Ablation studies

In this section, we perform two specific ablation studies: one at the model architecture level and another at the data level.

4.4.1 Ablation study at model architecture level

This ablation study consists of evaluating D2M performance in different configurations, including:

- **Without (w/o) decay factors**: the first admission was not weighted (see Equation 8), and the decay factors were not integrated into the information transfer score matrix (see Equation 10);
- **w/o similarity scores**: similarity scores (see Equations 3 and 5) were not integrated into the information transfer score matrix (see Equation 10);

The modified mathematical formulas for these different configurations and the results obtained are presented in Table 3. We observe a drop in D2M performance when decay factors, and similarity scores, are not used. These results reinforce our assertion of the importance of exploiting the time elapsed between admissions and their mutual similarities. We note that D2M performance is more affected when the decay factors matrix is excluded than when similarity scores are. We assume this is because, even without similarity scores, the decay factor matrix can implicitly capture how similar two admissions are based on the time elapsed. We believe that in a classification or regression task exclusively related to chronic diseases, D2M would perform better because the selection mask (see Equation 1) will be less sparse and therefore bring more information into the decision process.

Table 3: Ablation study at model architecture level.

Configuration	AUC	AUPRC	Formula
w/o decay factors	0.692 ± 0.012	0.359 ± 0.012	$Q^j = V^j$ $\Gamma^j = tanh([\phi^j \parallel \omega^j] W_\Gamma + B_\Gamma)$
w/o similarity scores	0.699 ± 0.012	0.365 ± 0.010	$\Gamma^j = tanh(\Delta^j W_\Gamma + B_\Gamma)$
Full	$\mathbf{0.705 \pm 0.007}$	$\mathbf{0.372 \pm 0.012}$	

± Standard deviation. w/o Without.

4.4.2 Ablation study at data level

We dedicate this second ablation study to assessing D2M performance when:

- Clinical codes are excluded: **w/o clinical codes**. We did not use clinical codes as input features (see Equation 14). Only complementary features are used;
- Complementary features (see subsection 3.1) are excluded: **w/o complementary features**.

Table 4: Ablation study at data level.

Input	AUC	AUPRC	Formula
w/o complementary features	0.683 ± 0.010	0.334 ± 0.013	$\hat{y} = g^1_{\beta_1} \circ g^2_{\beta_2} \cdots \circ g^S_{\beta_S}(\mathbf{q})$
w/o clinical codes	0.531 ± 0.002	0.183 ± 0.013	$\hat{y} = g^1_{\beta_1} \circ g^2_{\beta_2} \cdots \circ g^S_{\beta_S}(\tilde{\mathbf{d}})$
Full	$\mathbf{0.705 \pm 0.007}$	$\mathbf{0.372 \pm 0.012}$	

± Standard deviation.

The results reported in Table 4 show that D2M performs poorly when clinical codes and complementary features are not simultaneously used as inputs. We can conclude that the model performs better when some data are combined than when they are processed separately. For instance, breast cancer (diagnosis) and female (sex) will provide more information to the model when processed together rather than separately.

We have found that D2M performs better when it is fed only with clinical codes than when it is fed exclusively with complementary features. Obviously, we expected a significant performance drop by only exploiting the complementary features. The goal of this configuration was to show the contribution of clinical code data. This comparison leads us to deduce that diagnoses, prescriptions, and procedures (clinical codes) are the primary features that should be considered for unplanned hospital readmission prediction.

4.5 Explainability

In sensitive fields such as medicine, model accuracy is not the only factor that matters. The model must also be trustworthy. In other words, the model must be accurate and explainable. Not only is this important from a legal point of view, but it also enables healthcare professionals to understand the model's behaviour. The more explainable the model, the higher the likelihood of its adoption. To meet the second condition, i.e. explainability, we provide in this section several graphs that healthcare professionals can use to understand the factors driving the model's decisions. The explanation given through the graphs is based on clinical code embeddings (Equation 2), the decay factors (Equation 6), and the information transfer scores (Equation 10). The following paragraphs explore the reasoning behind the model based on a use case. Note that the words disease and diagnosis are used interchangeably in what follows.

Initially, to predict unplanned readmissions, patients' clinical codes need to be mapped to embedded vectors for efficient information representation. The first graph in Figure 3 shows the diseases (also called diagnoses) diagnosed at the patient's first admission in a two-dimensional embedding space. The values of the x and y axes cannot be interpreted; they are just mathematical representations of the data. However, we would expect similar diseases with a similar context closer together. Health professionals can use this first graph to study hidden relationships between different diseases and check whether the extracted information matches their knowledge.

In the second graph (Figure 4), time information, via the decay factors, is added to the disease embeddings (or diagnosis embeddings) of the patient's first admission. Decay factors allow the model to learn how long diseases last. The higher they are, the more likely it is that the diseases will still be present after some time and increase the risk of readmission. The lower they are, the less likely it is that the diseases will be present after some time, which could decrease the chance of readmission. In other words, the decay factor quantifies the presence of a disease over time. Clinical codes with a low decay factor should have a low impact on patient readmission. As an example, Primary open-angle glaucoma has a high decay factor in Figure 4, which makes sense because the disease is chronic and not curable. Also, a high decay factor of infection with drug-resistant microorganisms makes sense because it often requires a protracted treatment and increases the risk of complications, especially in the elderly population. Interesting is the low decay factor of anaplastic large cell lymphoma, which is a rare type of cancer which is fast-growing and often returns, compared to glaucoma.

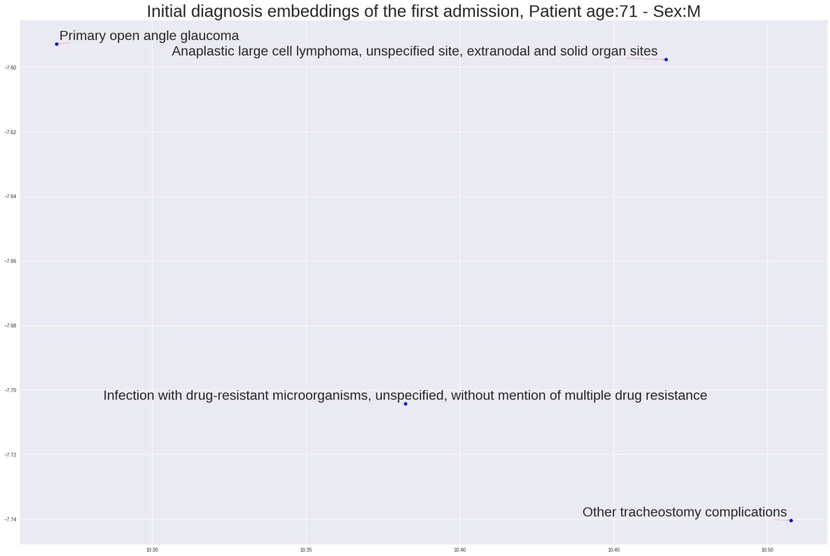


Fig. 3: Diagnosis embeddings of the patient's first admission.

This could be a hallucination of the model. Or the model really learned the fact that drug-resistant infections are often hard to treat in the elderly population and increase the risk of complications or even death. Overall, healthcare professionals will use the second graph to see how an illness lasts over time and to check whether this duration matches their knowledge.

The third graph (Figure 5) plays the same role as the first, except that the disease embeddings are those from the patient's second admission. In addition to the disease embeddings, we introduced information transfer scores in Figure 6 that quantify the amount of information added or subtracted from current diseases. This score is only available for diseases of the second admission that have been diagnosed during the first hospital admission (highlighted in yellow in Figures 5 and 6). The underlying assumption is that not all diseases are equally important for unplanned readmissions because there are common diseases, such as glaucoma with a lower impact, and more complex diseases such as malignancies. The higher the information transfer score, the more information on a disease that occurred during the previous admission is added or subtracted [5] to the information relating to the same disease during the second admission. This can be seen as a doctor merging information about the progress of the disease based on medical records from previous admissions in which the disease was diagnosed. This strategy enables the model to focus on chronic diseases and capture their evolution over time. This principle is shown in our example case (Figure 6) where

---

[5] Depending on the sign of the information transfer score.

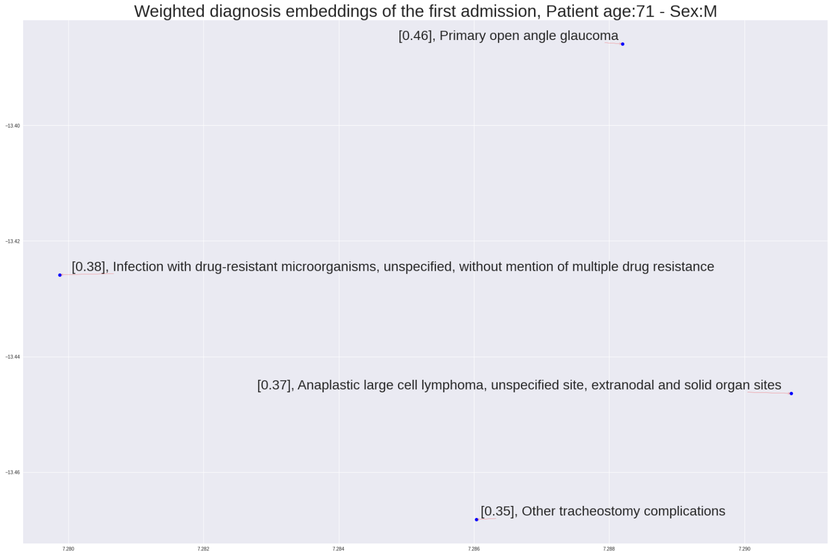


Fig. 4: Diagnosis embeddings of the patient's first admission with decay factors.

tracheostomy complications and large cell lymphoma scored higher because they are highly relevant for potential readmission. Healthcare professionals can use this fourth graph (Figure 6) to analyse the significance of diseases across successive admissions and check whether the information extracted aligns with their knowledge.

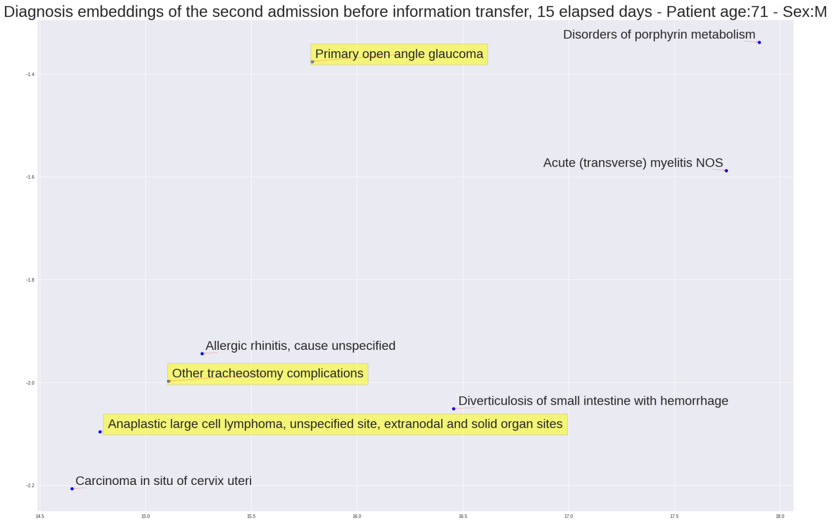


Fig. 5: Diagnosis embeddings of the patient's second admission.

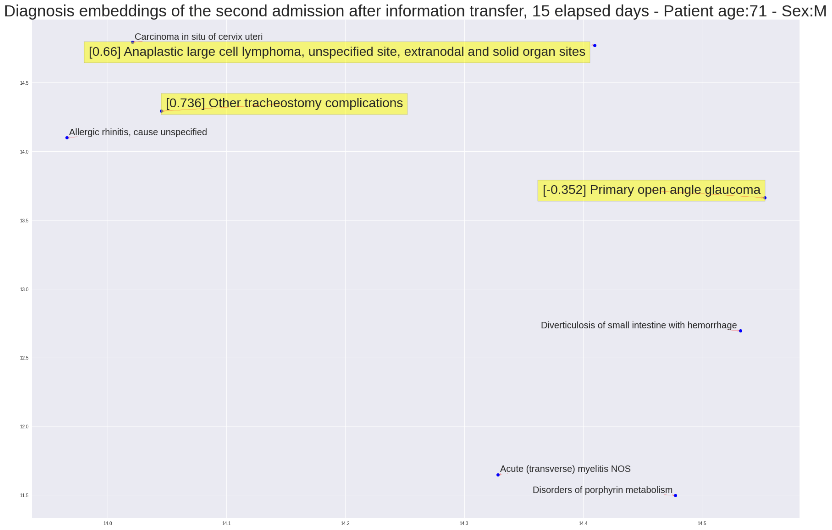


Fig. 6: Diagnosis embeddings of the patient's second admission with information transfer scores.

5 Discussion

Beyond its ability to deliver superior performance over competing models, it is worth mentioning that D2M has the advantage of being designed for any type of patient whose number of admissions may vary. This makes it an ideal candidate for use in real-life situations where patient medical profiles are heterogeneous. Moreover, in a world where climate change is a major concern, its relatively small number of parameters compared with existing models designed to solve the problem of predicting readmissions gives it an advantage in terms of energy and resource consumption.

While the above aspect makes D2M an ideal model for predicting unplanned hospital readmission, its explanatory capacity remains limited. Indeed, as described in subsection 4.5, through the decay factor value and information transfer score, we can quantify to what extent a diagnosis influences the prediction. However, these decay factor values and information transfer scores are not available for diagnoses in the current admission that do not appear in the most recent historical medical event. This is a limitation in terms of explainability, since the prediction may depend on these diagnoses. The priority for further work is then to find a way to assign weights to these diagnoses to make the D2M explainability capacity more complete.

As demonstrated in the figures in Section 4.5, the model might not capture current medical knowledge perfectly. We can observe some potential hallucinations in terms of the clustering of similar diseases, the decay factors, and the information transfer scores. Apart from hallucinations, there is also the possibility that the model captured and visualized connections that we have not yet understood and therefore cannot be explained using current medical knowledge. As is widely known in the data science community, this is typical for deep learning but needs to be taken into account and explained when demonstrating the capabilities of the models to healthcare professionals, where the knowledge of the technical functionality of deep learning might not be as high.

In terms of generalization, we assume that as long as some patients involved in a health study (classification, regression) have historical medical events with their corresponding registration dates; clinical codes can be encoded as categorical features (e.g. ICD code for diagnoses); and patient demographics and additional information are available, the proposed model can be used and can provide satisfactory results. However, we are aware of the need to evaluate the model with non-ICD codes with a coarser granularity than ICD. Indeed, there is no guarantee that the D2M performance is not due to the ability of ICD codes to encode a considerable amount of medical information at once.

Although unplanned readmissions have been used to quantify treatment quality, this approach brings certain limitations. With datasets that only contain data from one hospital, such as MIMIC, no unplanned readmission can also be the result of the patient being admitted to another treatment facility. Additionally, based on the health insurance system in the US, readmission might not happen because of financial issues. Another limitation from a clinical point of view is the use of ICD-9 codes. ICD-9 was retired in 2015. Disease classifications change over time. Compared to ICD-11, the current coding standard for diseases in healthcare, ICD-9 offers a less granular capturing of diseases and does not always reflect the current state of the art in medical knowledge. From a sampling perspective, the MIMIC dataset is primarily limited to ICU patients who typically have more severe conditions and consequently have a different treatment trajectory than other patients. That information is not captured in the data and therefore could introduce a bias in the prediction. Nevertheless, the MIMIC dataset has become a standard benchmark dataset for evaluating AI algorithms in medicine and has therefore been used for this study.

Even though the AUC is acceptable, the AUPRC of our current model is rather low, which is quite common also in similar studies [10, 44]. Although we outperform other model architectures, the AUPRC would need to be drastically improved in order to really bring benefit to patients and clinicians in a real-world scenario. We assume, that the low AUPRC is because of the relatively low number of positive samples. In future research, we plan to exploit a dataset containing a larger number of positive samples and include additional features such as laboratory reports and vital signs to improve AUPRC and AUC. To better understand the implications of the above-mentioned limitations, further research, especially the evaluation of the proposed D2M model in a clinical trial, is encouraged.

6 Conclusion

To improve the prediction of unplanned hospital readmissions, we designed a sequential deep learning model called Deep Magnitude Management (D2M) that processes patient medical information recorded during successive admissions. In contrast with the existing sequential models designed to address the unplanned hospital readmission challenge, D2M has the particularity of processing patients' medical information while taking into account the date on which this information was recorded. Additionally, D2M incorporates an explicit information transfer mechanism that allows it to focus on frequent medical events such as chronic diseases. These two strategies, which aim to reduce the impact of medical events that occurred a long time ago and focus on frequent medical events such as chronic diseases that often have a high impact on unplanned readmissions, make it possible to predict readmissions more accurately. We support our assertion by comparing D2M's performance with that of state-of-the-art models. Explainability being crucial in the medical field, we also offer an explainability component based on four graphs that can be explored by healthcare professionals to understand how the model arrived at a given decision. Following further evaluation and training with more data, we believe D2M will help healthcare professionals address the challenges associated with unplanned readmissions, including better management of medical resources, improved care and reduced expenditure for patients and healthcare institutions.

Appendix A Competitor models' setting

The hyperparameters and training configurations of competing models are as follows:

- LR: its penalty norm is L2 (ridge regression). It was trained over 1000 iterations;

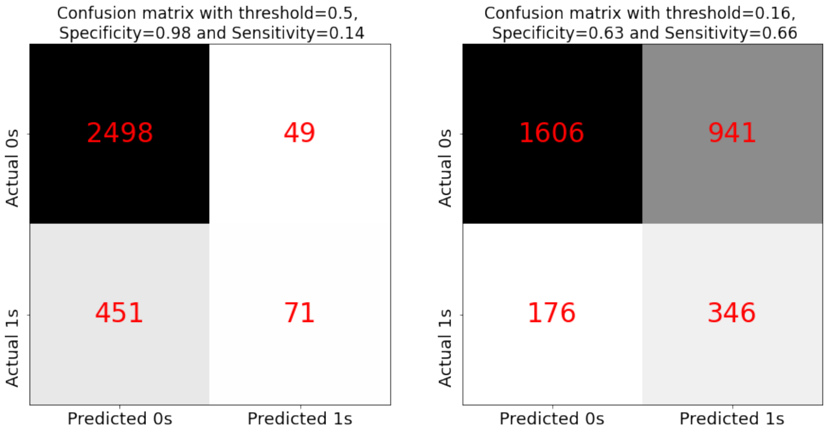
- LSTM [39]: the number of LSTM units is set to 300, and the training was done over 80 epochs;
- Timeline [28]: the dimension of the query and key vectors is set to 100, and the number of the two RNN units (because it is bidirectional) is set to 100. It was trained over 120 epochs;
- GRU-Decay [20]: the number of GRU units is set at 300 and they were trained over 170 epochs;
- Retain [40]: the number of its two RNN units (RNN_β and RNN_α) is set to 100 and it was trained over 40 epochs;
- Transformer [41]: the number of head attention is set to 4. Its feedforward neural network (FNN) has 100 units. A dropout of 0.8 is applied to the output of the FNN. It was trained over 50 epochs.

The remaining training hyperparameters are the same as those used to train our model.

Appendix B Confusion matrices

In this section, we present the confusion matrices for each model obtained from the training set of 1-fold cross-validation. Each model is evaluated with the default threshold 0.5 and the optimal threshold obtained with the Threshold-Moving technique. Sensitivity and Specificity are also reported.

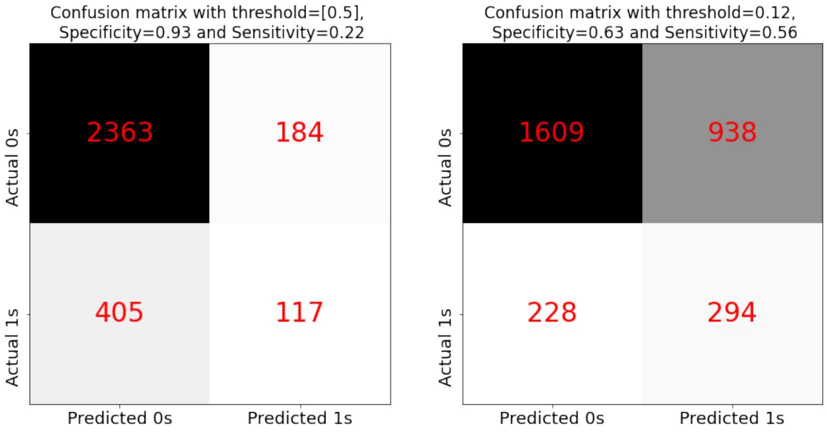
Although these confusion matrices do not reflect the overall performance of the models, they provide insight into the performance of each model.



Confusion matrix with threshold=0.5, Specificity=0.98 and Sensitivity=0.14

Confusion matrix with threshold=0.16, Specificity=0.63 and Sensitivity=0.66

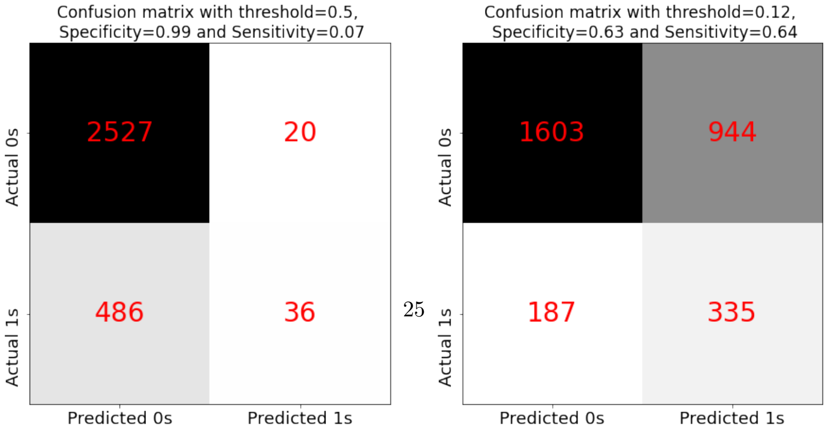
(a) GRU-Decay



Confusion matrix with threshold=[0.5], Specificity=0.93 and Sensitivity=0.22

Confusion matrix with threshold=0.12, Specificity=0.63 and Sensitivity=0.56

(b) LR

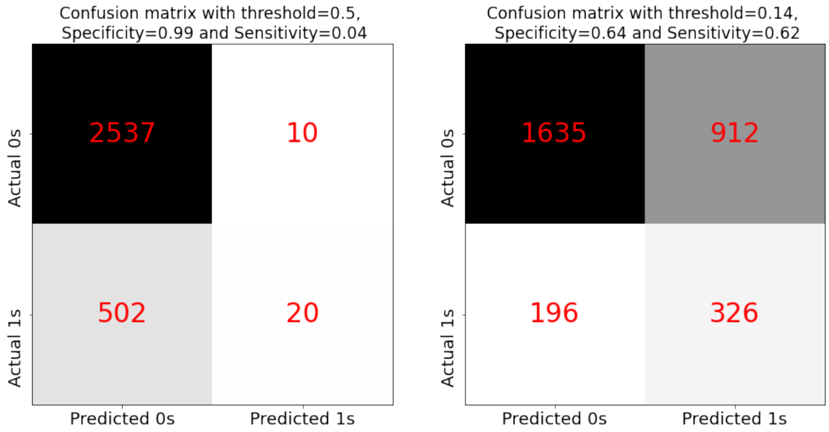


Confusion matrix with threshold=0.5, Specificity=0.99 and Sensitivity=0.07

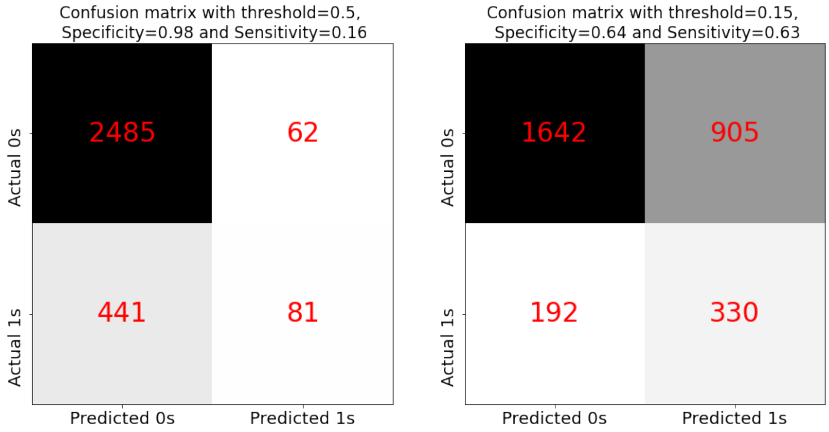
Confusion matrix with threshold=0.12, Specificity=0.63 and Sensitivity=0.64

(c) LSTM

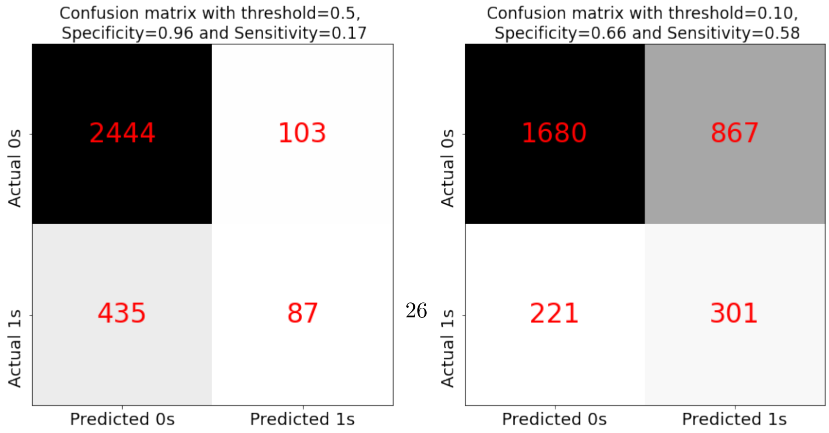
Fig. B1: Confusion matrices of GRU-Decay, LR and LSTM.



(a) Retain

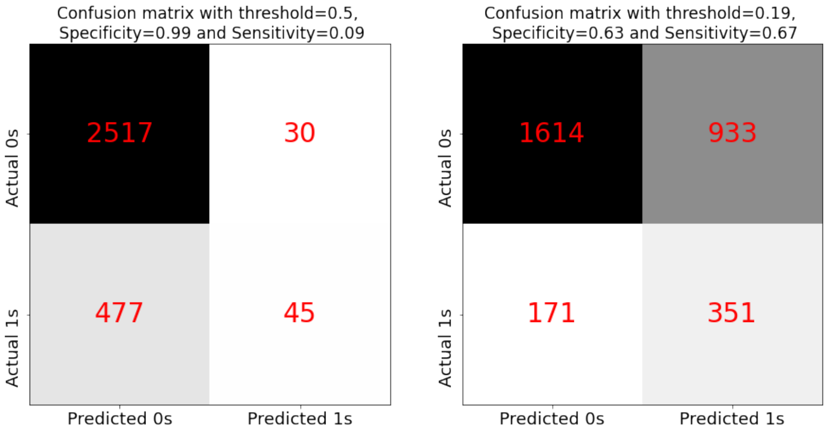


(b) Timeline



(c) Transformer

Fig. B2: Confusion matrices of Retain, Timeline and Transformer.



(a) D2M

Fig. B3: Confusion matrices of D2M.

References

[1] C. K. McIlvennan, Z. J. Eapen, L. A. Allen, Hospital readmissions reduction program, Circulation 131 (20) (2015) 1796–1803.

[2] A. K. Khanna, M. A. Moucharite, P. J. Benefield, R. Kaw, Patient characteristics and clinical and economic outcomes associated with unplanned medical and surgical intensive care unit admissions: A retrospective analysis, ClinicoEconomics and Outcomes Research (2023) 703–719.

[3] H. C. Felix, B. Seaberg, Z. Bursac, J. Thostenson, M. K. Stewart, Why do patients keep coming back? results of a readmitted patient survey, Social work in health care 54 (1) (2015) 1–15.

[4] H. Hyppönen, J. Viitanen, J. Reponen, P. Doupi, V. Jormanainen, T. Lääveri, J. Vänskä, I. Winblad, P. Hämäläinen, Large-scale ehealth systems: providing information to support evidence-based management, in: The Third International Conference on eHealth, Telemedicine, and Social Medicine. February 23-28, 2011-Gosier, Guadeloupe, France, IARIA XPS Press, 2011, pp. 89–95.

[5] S. Basu Roy, A. Teredesai, K. Zolfaghar, R. Liu, D. Hazel, S. Newman, A. Marinez, Dynamic hierarchical classification for patient risk-of-readmission, in: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 1691–1700.

[6] P. Flach, Machine learning: the art and science of algorithms that make sense of data, Cambridge university press, 2012.

[7] T. Katsuki, K. Miyaguchi, A. Koseki, T. Iwamori, R. Yanagiya, A. Suzuki, Cumulative stay-time representation for electronic health records in medical event time

prediction, in: L. D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, ijcai.org, 2022, pp. 3861–3867. `doi:10.24963/IJCAI.2022/536`.
URL https://doi.org/10.24963/ijcai.2022/536

[8] Y. Lee, E. Jun, H. Suk, Multi-view integration learning for irregularly-sampled clinical time series, CoRR abs/2101.09986 (2021). `arXiv:2101.09986`.
URL https://arxiv.org/abs/2101.09986

[9] M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, D. Draheim, Evaluation of deep learning-based depression detection using medical claims data, Artificial Intelligence in Medicine 147 (2024).

[10] C. Xiao, T. Ma, A. B. Dieng, D. M. Blei, F. Wang, Readmission prediction via deep contextual embedding of clinical concepts, PloS one 13 (4) (2018) e0195024.

[11] A. Ashfaq, A. Sant'Anna, M. Lingman, S. Nowaczyk, Readmission prediction using deep learning on electronic health records, Journal of biomedical informatics 97 (2019) 103256.

[12] C.-Y. Chi, S. Ao, A. Winkler, K.-C. Fu, J. Xu, Y.-L. Ho, C.-H. Huang, R. Soltani, et al., Predicting the mortality and readmission of in-hospital cardiac arrest patients with electronic health records: A machine learning approach, Journal of medical Internet research 23 (9) (2021) e27798.

[13] K. Huang, J. Altosaar, R. Ranganath, Clinicalbert: Modeling clinical notes and predicting hospital readmission, CoRR abs/1904.05342 (2019). `arXiv:1904.05342`.
URL http://arxiv.org/abs/1904.05342

[14] A. S. Imran, S. M. Daudpota, Z. Kastrati, R. Batra, Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets, IEEE Access 8 (2020) 181074–181090. `doi:10.1109/ACCESS.2020.3027350`.
URL https://doi.org/10.1109/ACCESS.2020.3027350

[15] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, L. S. Chao, Learning deep transformer models for machine translation, in: A. Korhonen, D. R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 1810–1822. `doi:10.18653/V1/P19-1176`.
URL https://doi.org/10.18653/v1/p19-1176

[16] Y. Li, M. Mamouei, G. Salimi-Khorshidi, S. Rao, A. Hassaïne, D. Canoy, T. Lukasiewicz, K. Rahimi, Hi-behrt: Hierarchical transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records, IEEE J. Biomed. Health Informatics 27 (2) (2023) 1106–1117. `doi:10.1109/JBHI.2022.3224727`.
URL https://doi.org/10.1109/JBHI.2022.3224727

[17] M. D. Naylor, D. Brooten, R. Campbell, B. S. Jacobsen, M. D. Mezey, M. V. Pauly, J. S. Schwartz, Comprehensive discharge planning and home follow-up of hospitalized elders: a randomized clinical trial, Jama 281 (7) (1999) 613–620.

[18] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R. G. Mark, Mimic-iii, a freely accessible critical care database, Scientific data 3 (1) (2016) 1–9.

[19] N. Arya, S. Saha, Multi-modal advanced deep learning architectures for breast cancer survival prediction, Knowl. Based Syst. 221 (2021) 106965. `doi:10.1016/J.KNOSYS.2021.106965`.
URL https://doi.org/10.1016/j.knosys.2021.106965

[20] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Scientific reports 8 (1) (2018) 6085.

[21] S. Jiang, K. Chin, G. Qu, K. Tsui, An integrated machine learning framework for hospital readmission prediction, Knowl. Based Syst. 146 (2018) 73–90. `doi:10.1016/J.KNOSYS.2018.01.027`.
URL https://doi.org/10.1016/j.knosys.2018.01.027

[22] X. Liu, Y. Chen, J. Bae, H. Li, J. Johnston, T. Sanger, Predicting heart failure readmission from clinical notes using deep learning, in: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 2642–2648.

[23] N. B. Thapa, S. Seifollahi, S. Taheri, Hospital readmission prediction using clinical admission notes, in: Australasian Computer Science Week 2022, 2022, pp. 193–199.

[24] M. Bertl, K. J. I. Kankainen, G. Piho, D. Draheim, P. Ross, Evaluation of Data Quality in the Estonia National Health Information System for Digital Decision Support, in: Proceedings of the 3rd International Health Data Workshop, CEUR-WS, 2023.

[25] A. Hammoudeh, G. Al-Naymat, I. Ghannam, N. Obied, Predicting hospital readmission among diabetics using deep learning, Procedia Computer Science 141 (2018) 484–489.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[27] X. Min, B. Yu, F. Wang, Predictive modeling of the hospital readmission risk from patients' claims data using machine learning: a case study on copd, Scientific reports 9 (1) (2019) 1–10.

[28] T. Bai, S. Zhang, B. L. Egleston, S. Vucetic, Interpretable representation learning for healthcare via capturing disease progression through time, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 43–51.

[29] C. A. Parker, N. Liu, S. X. Wu, Y. Shen, S. S. W. Lam, M. E. H. Ong, Predicting hospital admission at the emergency department triage: A novel prediction model, The American journal of emergency medicine 37 (8) (2019) 1498–1504.

[30] M. Pishgar, J. Theis, M. Del Rios, A. Ardati, H. Anahideh, H. Darabi, Prediction of unplanned 30-day readmission for icu patients with heart failure, BMC medical informatics and decision making 22 (1) (2022) 117.

[31] R. M. Carvalho, D. Oliveira, C. Pesquita, Knowledge graph embeddings for icu readmission prediction, BMC Medical Informatics and Decision Making 23 (1) (2023) 12.

[32] E. Choi, Z. Xu, Y. Li, M. Dusenberry, G. Flores, E. Xue, A. Dai, Learning the graphical structure of electronic health records with graph convolutional transformer, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 606–613.

[33] C. A. Low, D. H. Bovbjerg, S. Ahrendt, M. H. Choudry, M. Holtzman, H. L. Jones, J. F. Pingpank Jr, L. Ramalingam, H. J. Zeh III, A. H. Zureikat, et al., Fitbit step counts during inpatient recovery from cancer surgery as a predictor of readmission, Annals of Behavioral Medicine 52 (1) (2018) 88–92.

[34] A. Doryab, A. K. Dey, G. Kao, C. Low, Modeling biobehavioral rhythms with passive sensing in the wild: a case study to predict readmission risk after pancreatic surgery, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (1) (2019) 1–21.

[35] C. Qian, P. Leelaprachakul, M. Landers, C. Low, A. K. Dey, A. Doryab, Prediction of hospital readmission from longitudinal mobile data streams, Sensors 21 (22) (2021) 7510.

[36] J. Stehlik, C. Schmalfuss, B. Bozkurt, J. Nativi-Nicolau, P. Wohlfahrt, S. Wegerich, K. Rose, R. Ray, R. Schofield, A. Deswal, et al., Continuous wearable monitoring analytics predict heart failure hospitalization: the link-hf multicenter study, Circulation: Heart Failure 13 (3) (2020) e006513.

[37] W. J. Kane, T. E. Hassinger, E. L. Myers, D. L. Chu, A. N. Charles, S. C. Hoang, C. M. Friel, R. H. Thiele, T. L. Hedrick, Wearable technology and the association of perioperative activity level with 30-day readmission among patients undergoing major colorectal surgery, Surgical Endoscopy 36 (2) (2022) 1584–1592.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[39] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[40] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, W. Stewart, Retain: An interpretable predictive model for healthcare using reverse time attention mechanism, Advances in neural information processing systems 29 (2016).

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[42] A. M. Tjoa, S. Tjoa, The role of ict to achieve the un sustainable development goals (sdg), in: ICT for Promoting Human Development and Protecting the Environment: 6th IFIP World Information Technology Forum, WITFOR 2016, San José, Costa Rica, September 12-14, 2016, Proceedings 6, Springer, 2016, pp. 3–13.

[43] S. Georgiou, M. Kechagia, T. Sharma, F. Sarro, Y. Zou, Green ai: Do deep learning frameworks have different costs?, in: Proceedings of the 44th International Conference on Software Engineering, 2022, pp. 1082–1094.

[44] Y. Huang, A. Talwar, Y. Lin, R. R. Aparasu, Machine learning methods to predict 30-day hospital readmission outcome among us adults with pneumonia: analysis of the national readmission database, BMC Medical Informatics and Decision Making 22 (1) (2022) 288.

Curriculum Vitae

Personal data

Name	Nzamba Bignoumba
Date and place of birth	8 April 1991 Libreville, Gabon
Nationality	Gabonese

Contact information

E-mail	nzamba.bignoumba@taltech.ee

Education

2020-2024	Tallinn University of Technology, School of Information Technologies, Computer Science, PhD studies
2017-2019	Faculty of Mathematics, Physics and Natural Science of Tunis Computer Science, MSc
2013-2016	African Institute of Computer Science Computer Science, Programmer Analyst (Project Engineer)

Language competence

French	native
English	fluent

Professional employment

2016-2017	Fontecsys, Graphic designer

Supervision (Defended)

- 2024, Pavel Grubelja, Transformer-based model for predicting hospital readmissions, BSc, supervisor Prof. Sadok Ben Yahia, **Nzamba Bignoumba**, Tallinn University of Technology.
- 2021, Lavender Amondi, Data governance and access to healthcare in kenya; a survey of government hospitals in nairobi county, MSc, supervisor Prof. Sadok Ben Yahia, **Nzamba Bignoumba**, Tallinn University of Technology.

Scientific work

1. N. Bignoumba and S. Ben Yahiaand N. Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. In *Proceedings of KES'2024 - the 28th Annual KES Conference*, 2024
2. M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, and D. Draheim. Evaluation of deep learning-based depression detection using medical claims data. *Artificial Intelligence in Medicine*, 147:102745, 2024
3. N. Bignoumba, N. Mellouli, and S. B. Yahia. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. *Expert Systems with Applications*, 238:122148, 2024
4. N. Bignoumba, S. B. Yahia, and N. Mellouli. Étude de similarité des patients pour identifier les unités hospitalières ayant le taux le plus élevé de réadmissions non planifiées. *Actes de la journée d'étude sur la Similarité entre Patients, SimPa 2023*, page 24, 2023
5. S. Elloumi and N. Bignoumba. Stacking deep-learning model, stories and drawing properties for automatic scene generation. *International Journal of Advanced Computer Science and Applications*, 14(1), 2023

Project participation

2020-2021	Eurora Project. The STACC and Tallinn University of Technology Data scientist, Models generation for HS-Code classification.

Conference presentations

1. N. Bignoumba, S. B. Yahia, and N. Mellouli. *Étude de similarité des patients pour identifier les unités hospitalières ayant le taux le plus élevé de réadmissions non planifiées*, Simpa: 13 March, 2023, Paris

Elulookirjeldus

Isikuandmed

Nimi	Nzamba Bignoumba
Sünniaeg ja -koht	8. aprill 1991 Libreville, Gabon
Kodakondsus	Gaboni

Kontaktandmed

E-post	nzamba.bignoumba@taltech.ee

Haridus

2020-2024	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond Arvutiteaduse doktoriõpingud
2017-2019	Tunise matemaatika-, füüsika- ja loodusteaduste teaduskond Arvutiteaduse magister
2013-2016	Aafrika arvutiteaduste instituut Arvutiteadus, programmeerija analüütik (projektiinsener)

Keelteoskus

Prantsuse	emakeel
Inglise	vabalt valdav

Töökogemus

2016-2017	Fontecsys, Graafiline disainer

Juhendamine (kaitsmised)

- 2024, Pavel Grubelja, Haiglasse tagasipöördumiste ennustamiseks põhinev transformer mudel, bakalaureus, juhendaja Prof. Sadok Ben Yahia, **Nzamba Bignoumba**, Tallinna Tehnikaülikool.

- 2021, Lavender Amondi, Andmete haldamine ja juurdepääs tervishoiuteenustele keenias; valitsushaiglate uuring nairobi maakonnas , magister, juhendaja Prof. Sadok Ben Yahia, **Nzamba Bignoumba**, Tallinna Tehnikaülikool.

Publikatsioonid

- N. Bignoumba and S. Ben Yahiaand N. Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. In *Proceedings of KES'2024 - the 28th Annual KES Conference*, 2024
- M. Bertl, N. Bignoumba, P. Ross, S. B. Yahia, and D. Draheim. Evaluation of deep learning-based depression detection using medical claims data. *Artificial Intelligence in Medicine*, 147:102745, 2024
- N. Bignoumba, N. Mellouli, and S. B. Yahia. A new efficient alignment-driven neural network for mortality prediction from irregular multivariate time series data. *Expert Systems with Applications*, 238:122148, 2024
- N. Bignoumba, S. B. Yahia, and N. Mellouli. Étude de similarité des patients pour identifier les unités hospitalières ayant le taux le plus élevé de réadmissions non planifiées. *Actes de la journée d'étude sur la Similarité entre Patients, SimPa 2023*, page 24, 2023
- S. Elloumi and N. Bignoumba. Stacking deep-learning model, stories and drawing properties for automatic scene generation. *International Journal of Advanced Computer Science and Applications*, 14(1), 2023

Projektides osalemine

2020-2021	Eurora projekt. STACC ja Tallinna Tehnikaülikool. Andmeanalüütik, mudelite genereerimine HS-koodide klassifitseerimiseks.

Konverentsiettekanded

1. N. Bignoumba, S. B. Yahia, and N. Mellouli. *Étude de similarité des patients pour identifier les unités hospitalières ayant le taux le plus élevé de réadmissions non planifiées*, Simpa: 13 March, 2023, Paris

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-195-6 (PDF)