TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Pavel Grubeljas 206504IAIB

TRANSFORMER-BASED MODEL FOR PREDICTING HOSPITAL READMISSIONS

Bachelor's Thesis

Supervisor: Sadok Ben Yahia
PhD

Co-supervisor: Nzamba Bignoumba
MSc

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Pavel Grubeljas 206504IAIB

HAIGLASSE TAGASIPÖÖRDUMISTE ENNUSTAMISEKS PÕHINEV TRANSFORMER MUDEL

Bakalaureusetöö

Juhendaja: Sadok Ben Yahia
PhD

Kaasjuhendaja: Nzamba Bignoumba
MSc

Tallinn 2024

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Pavel Grubeljas

27.05.2024

Abstract

The landscape of predictive analytics in healthcare has been significantly transformed by the advent of deep learning methodologies. This study introduces a Transformer-based model, specifically designed to predict hospital readmissions, and compares its performance against other state-of-the-art algorithms. The Transformer model leverages the textual data prevalent in clinical notes to capture nuanced patient information indicative of readmission risks. By utilizing self-attention mechanisms, the model comprehensively assimilates context from long sequences of clinical narratives, enabling more informed predictions. The time information is handled by a time decay factor, emphasizing the most recent admissions, which is a significant aspect of the model, hence the name Transformer+Decay. The model's architecture facilitates the encoding of temporal patient information, addressing the complexity of patient trajectories.

We benchmarked the Transformer model against widely-used algorithms such as LSTM and LR. Our comparative analysis was conducted on a dataset comprising diverse patient encounters, evaluated using metrics such as AUC and AUPRC. The Transformer model demonstrated superior performance, indicating its potential as a robust tool for assisting healthcare providers in early intervention efforts. This study not only underscores the efficacy of Transformer models in a clinical setting but also highlights the importance of integrating narrative clinical data for enhancing predictive accuracy.

The goal of the work has been achieved. Transformer+Decay showed the best result in AUPRC compared to other models. For improving AUC, further research is needed.

The thesis is written in English and is 32 pages long, including 4 chapters, 3 figures and 5 tables.

Annotatsioon

Ennustavate analüütikate maastik tervishoius on süvaõppemeetodite kasutuselevõtuga märkimisväärselt muutunud. Käesolev uurimus tutvustab haigla taashospidaliseerimise ennustamiseks spetsiaalselt välja töötatud transformer-mudelit ja võrdleb selle jõudlust teiste tipptasemel algoritmidega. Transformer-mudel kasutab kliinilistes märkmetes leiduvaid tekstilisi andmeid, et tabada taashospidaliseerimise riski näitavaid nüansirikkaid patsiendiandmeid. Enese tähelepanu mehhanismide abil koondab mudel põhjalikult konteksti pikkadest kliiniliste narratiivide järjestustest, võimaldades teadlikumaid ennustusi. Aja teavet käsitletakse ajakulu teguriga, rõhutades kõige hiljutisemaid hospitaliseerimisi, mis on mudeli oluline aspekt ja seetõttu nimetatakse seda Transformer+Decay. Mudeli arhitektuur võimaldab kodeerida ajalisi patsiendiandmeid, käsitledes patsiendi trajektooride keerukust.

Võrdlesime transformer-mudelit laialdaselt kasutatavate algoritmidega nagu LSTM ja LR. Meie võrdlev analüüs viidi läbi andmekogumi põhjal, mis sisaldas mitmekesiseid patsientide kohtumisi ja mida hinnati selliste mõõdikute abil nagu AUC ja AUPRC. Transformer-mudel näitas paremat jõudlust, mis viitab selle potentsiaalile olla usaldusväärne tööriist tervishoiuteenuste osutajatele varajase sekkumise jõupingutuste toetamisel. See uurimus rõhutab mitte ainult transformer-mudelite tõhusust kliinilises keskkonnas, vaid ka narratiivsete kliiniliste andmete integreerimise tähtsust ennustustäpsuse suurendamisel.

Töö eesmärk on saavutatud. Transformer+Decay näitas teiste mudelitega võrreldes parimat tulemust AUPRC-s. AUC parandamiseks on vaja edasisi uuringuid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 32 leheküljel, 4 peatükki, 3 joonist, 5 tabelit.

List of Abbreviations and Terms

NLP	Natural Language Procssing
EHR	Electronic Health Record
MIMIC	Medical Information Mart for Intensive Care
ICU	Intensive Care Unit
FNN	Feedforward Neural Networks
GRU	Gated Recurrent Unit
CNN	Convolutional Neural Networks
ROC	Receiver Operating Characteristic Curve
AUC	Area Under the Receiver Operating Characteristic curve
LSTM	Long Short-Term Memory
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
TPR	True Positive Rate
FPR	False Positive Rate
ReLU	Rectified Linear Function

Table of Contents

List of Figures

List of Tables

1. Introduction

Hospital readmission is the process by which patients return to the hospital for further treatment or care within a specific time frame after their initial discharge. Hospital readmissions are a major challenge for healthcare systems worldwide, as they can result in increased costs, reduced patient outcomes, and a strain on hospital resources [1].

In addition to the financial burden, readmissions can also have a significant impact on patient health and quality of life, as they may indicate that the initial care provided was inadequate or that the patient has underlying health issues that need to be addressed. Hospital readmission prediction has garnered significant attention due to its implications for managing emergency department overcrowding [1]. Various studies have explored this critical area using deep learning techniques, leveraging diverse datasets and models to enhance predictive accuracy.

Predicting which patients are at high risk of readmission can help healthcare providers to intervene early and prevent unnecessary readmissions. EHRs contain a wealth of information that can be used to build predictive models for readmission risk. EHRs typically include data on patients' demographics, medical history, medication usage, and diagnostic codes, among other things. However, the complexity and variability of this data require sophisticated machine learning techniques to extract meaningful patterns and make accurate predictions.

1.1 Goals

This thesis aims to advance the field of medical informatics by exploring the application of transformer-based models for hospital readmission prediction and comparing their performance with classical machine learning architectures. Our transformer-based model is a variant of the original one proposed in [2]. Indeed, the original Transformer was designed for a NLP task. Therefore, to adapt it to medical tasks where time plays an important role, we added a layer called the decay factor layer dedicated to encoding the irregular time elapsed between sequences of medical events. The goal can be segmented into the following subtasks:

- Preparing and parsing the dataset, which includes heterogeneous data (clinical codes and patient demographics);

- Design a decay factor layer dedicated to modelling time between visits;
- Performing end-to-end training of the decay factor layer with the transformer for mutual optimization of parameters;
- Comparison of our model with classical machine learning models designed for sequential modelling using the MIMIC-3 database;
- Carrying out an analysis of the results.

1.2 Related Works

Across the literature, a consistent focus emerges on predicting readmissions within a 30-day time frame [1-7]. Noteworthy contributions include the utilization of various predictive models and datasets tailored to specific medical conditions. For instance, in one study [3], a diverse set of machine learning models versus a deep learning model namely multi layer perceptron (MLP) were applied to predict readmissions in patients with Chronic Obstructive Pulmonary Disease (COPD), using a larger dataset comprising 111,992 medical records. They concluded throughout extensive empirical experiments that the Gradient Boosting Decision Trees (GBDT) yields optimal results compared to MLP, which is more sophisticated. Another study, in [4], also compared various machine learning models to MLP on unplanned readmission for ICU patients with heart failure. In contrast to [3], this study shown that MLP outperform machine learning models. As medical data is often organized longitudinally, several studies have leveraged the RNN and its variants, calling LSTM and GRU), to enhance the prediction of unplanned readmissions.

In the study [5], the LSTM model was employed to predict readmissions in patients with heart attack failure, using a relatively modest dataset of 7500 medical records. Remarkably, this study achieved a commendable AUC of 0.83, demonstrating the efficacy of the LSTM model in this context. The authors, in [6], also trained an RNN, GRU, and LSTM with an attention mechanism to predict unplanned ICU readmission from patient physiological time series. Thanks to the attention mechanism, they were able to provide accurate and explainable results. Similar work combining LSTM and attention mechanism was also carried out in [7] to exclusively predict readmission of breast cancer patients. Another study [8] focused on predicting unplanned readmission of heart failure patients and employed a combination of contextual embedding and RNN models. Despite utilising a smaller dataset, this study achieved functional results.

Various non- or partially RNN models were also applied to the readmission prediction task. For instance, the authors in [9] utilised CNN model to predict readmissions among diabetic patients, leveraging a substantial dataset of 100,000 medical records. They achieved remarkable performance that could be attributed to the complexity and richness inherent

in diabetic patient data, which may contain diverse and informative features conducive to accurate prediction. In [10], in order to improve the accuracy of the ICU readmission prediction task, the authors used biomedical ontologies to develop knowledge graphs aimed at improving the semantic extraction of clinical features and obtaining better encoding of their relationship. Another investigation, in [11], adopted a hierarchical vectoriser (HVec) deep learning model to predict readmissions and mortality, utilizing a substantial dataset of 256,589 readmission records. Notably, despite not focusing on a specific medical condition, this study achieved impressive performance, highlighting the effectiveness of the chosen modelling approach.

In a different approach, a study [12] employed the BERT model on a medium-sized dataset of 58,976 admissions, without specific focus on a particular medical condition. Despite the moderate dataset size, this study achieved a respectable AUC of 0.714, showcasing the efficacy of the transformer-based model in healthcare prediction tasks. In a pioneering effort, researchers in a recent study [13] utilized a modern language model, specifically an attention-based Transformer with BERT architecture, for readmission prediction. By harnessing the power of advanced language processing techniques, the study achieved notable success in accurately predicting hospital readmissions. Another notable endeavor [14] utilized a modified version of the transformer model, namely the Multimodal Spatiotemporal Graph-Transformer. This model integrated diverse data inputs, including EHR, X-ray images, and medical notes. Despite the complexity of integrating multimodal data sources, the study reported promising results, demonstrating the efficacy of transformer-based models in handling heterogeneous healthcare data. Similarly, researchers in another study [15] employed a multimodal transformer architecture for mortality prediction tasks. This approach involved incorporating clinical notes and time series information into the model, enabling a comprehensive analysis of patient data. Despite the additional complexity introduced by multimodal data integration, the study yielded encouraging results, further highlighting the versatility of transformer-based models in healthcare analytics.

In summary, the majority of researchers in the field of hospital readmission prediction have focused on developing predictive models tailored to specific medical conditions—a practice that, while valuable for understanding condition-specific risk factors, may limit the broader applicability of the models. Consequently, there is a growing need for the development of a universal readmission prediction model capable of accurately predicting readmission risk across diverse patient populations and medical conditions. To build upon these findings, our study aim is to conduct a comparative analysis, pitting a customized Transformer against various neural networks. By including benchmark models, such as LSTMs and FFNs, we seek to discern the strengths and weaknesses of each approach in the specific challenge of hospital readmission prediction. We will prioritize AUC as the

primary metric for evaluating model performance, while ensuring that the dataset size is comparable to the medium. Another key feature of our model will be a time decay layer to enhance the relevance of admissions in the predictive model.

2. Method

In this section will be discussed elements of transformer-decay model and how it works.

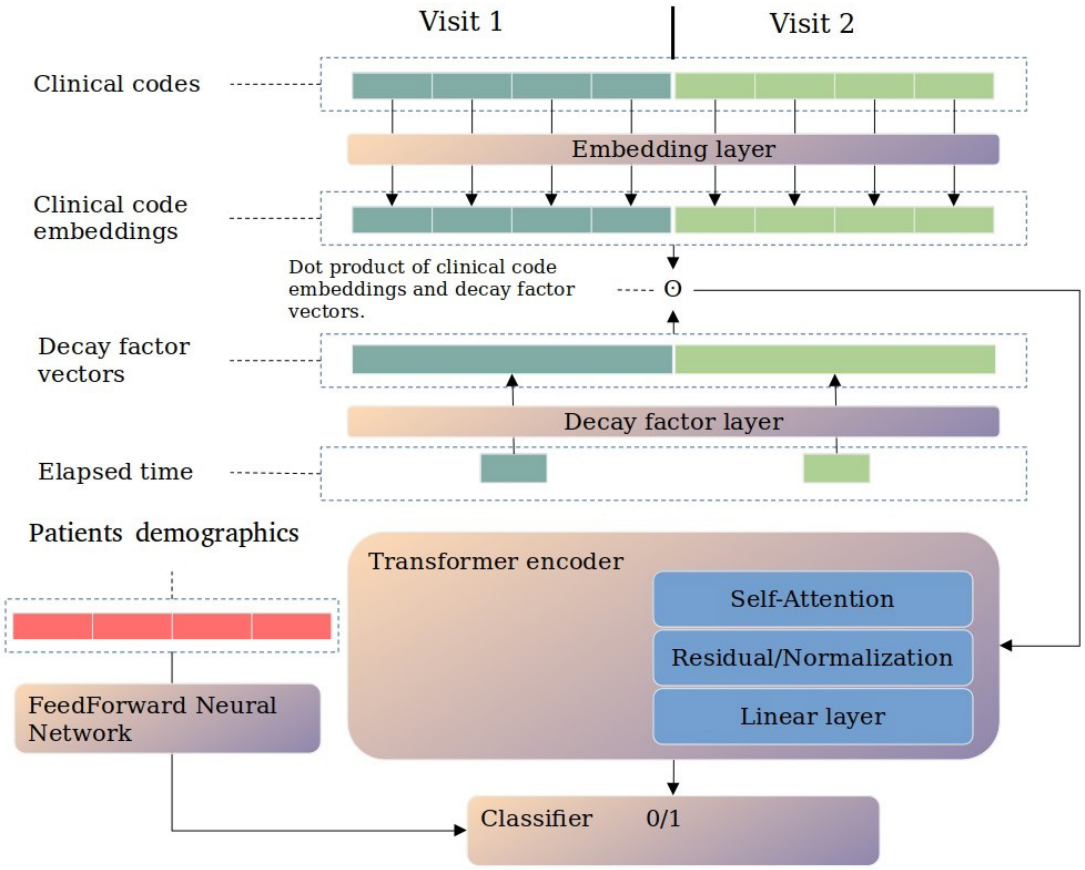

Figure 1. *Model architecture.*

2.1 K-fold cross-validation

K-fold cross-validation is a robust statistical method used in machine learning and data science to evaluate the performance of a predictive model. The dataset is randomly divided into k equally (or nearly equally) sized subsets or "folds". The model is trained k times, each time using a different fold as the validation set and the remaining $k-1$ folds as the training set. This means that each fold gets a chance to be the validation set exactly once. After each of the k iterations, the performance metric (e.g., accuracy, precision, recall, F1 score) is calculated and recorded. The final performance metric is obtained by averaging the k recorded performance metrics. This provides a more reliable estimate of the model's

performance compared to a single train-test split, as it reduces the variability associated with the particular division of the dataset.

By using multiple training and validation sets, k-fold cross-validation helps ensure the model's performance is not overly dependent on the specific partitioning of the data. All data points are used for both training and validation, maximizing the use of the available data. The average performance metric is generally more stable and less biased than metrics obtained from a single train-test split.

2.2 Classifier

The construction of a neural network for the prediction of patient readmissions necessitates a discerning approach to the selection of an activation function for the output layer, which acts as a binary classifier. The sigmoid function is employed for this purpose due to its pertinent properties for binary classification tasks.

Mathematically, the sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

where *e* is the mathematical constant, and *x* signifies the input to the function. This logistic function efficiently maps input values to a probabilistic range between 0 and 1.

2.3 Evaluation Metrics

In this section, we will discuss the main evaluation metrics that will be employed to describe the results of neural network models. For the explanation of metrics are used terms for classification model. Here is what each term represents:

1. True Positives (TP): The number of instances correctly predicted as positive.
2. True Negatives (TN): The number of instances correctly predicted as negative.
3. False Positives (FP): The number of instances incorrectly predicted as positive (also known as Type I error).
4. False Negatives (FN): The number of instances incorrectly predicted as negative (also known as Type II error).

2.3.1 Area under the ROC Curve (AUC)

The AUC is a performance metric for classification models at various threshold settings. The ROC curve is a graphical representation that plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels. The AUROC provides a single numerical metric that summarizes the model's ability to discriminate between the positive and negative classes over all possible thresholds. A higher AUROC indicates better model performance, with a value of 1 representing perfect classification and 0.5 denoting no discriminative power, equivalent to random guessing.

2.3.2 Area under Precision-Recall Curve (AUPRC)

The Area Under the Precision-Recall Curve is a metric used to evaluate the performance of a classification model, particularly in scenarios where there is a significant imbalance between classes. For better understanding it is better to explain what precision and recall are.

Precision

Precision is a metric used to measure a model's accuracy in predicting positive instances. It is calculated as the ratio of true positive predictions to the sum of positive predictions made (both true positives and false positives). High precision indicates that when the model predicts an instance as positive, it is likely correct.

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall is a metric used to measure the proportion of actual positive cases that a model correctly identifies. It is calculated as the ratio of true positives to the sum of true positives and false negatives. High recall indicates that the model is effective at capturing most of the relevant instances. The formula of recall can be stated as follows:

$$Recall = \frac{TP}{TP + FN}$$

The Precision-Recall Curve is a graph that plots Precision against Recall, at various threshold levels. The AUPRC provides a single value summarizing the trade-off between Precision and Recall across different thresholds. A higher AUPRC indicates a model that is

both accurate and sensitive, which is particularly valuable in cases where positive instances are rare or when the costs of False Negatives are high. Unlike the ROC curve which is affected by the large number of True Negatives in imbalanced datasets, the Precision-Recall Curve focuses on the minority class, making AUPRC a more appropriate metric in these situations.

2.4 Embedding layer

The embedding layer serves as a crucial component that translates both input and output tokens into vectors of a predetermined size, known as d_{model} [12]. This layer essentially functions as a sophisticated mapping tool, transforming each token into a dense vector that encapsulates its semantic significance and contextual relevance within the sequence. A distinctive feature of this setup is the sharing of the embedding layer between the model's input and output segments. This means that the same weight matrix is utilized not only for the embedding layer but also for the pre-softmax linear transformation that occurs later in the process. Additionally, it's important to note that within these embedding layers, the weights undergo scaling by the square root of d_{model} to maintain a balance in the magnitude of the input vectors. This approach ensures a more effective and nuanced representation of the tokens, which is vital for the model's performance in various language processing tasks.

2.5 Decay factor layer

The decay factor layer aims to map the time elapsed between two consecutive visits in a continuous vector. This vector will reduce the magnitude of clinical code embeddings that belong to a visit that took place a long time ago. Meanwhile, for those belonging to the most recent visits, their magnitude will remain almost unchanged. The underlying intuition is to avoid the model relying too heavily on historical and medical events that happened a long time ago, and to focus more on more recent medical events.

2.6 Transformer encoder

The Transformer encoder is a key component of the Transformer network architecture, which is used for sequence transduction tasks such as machine translation. The encoder is composed of a stack of N identical layers, each of which has two sub-layers. The first sub-layer is a multi-head self-attention mechanism, which allows each position in the sequence to attend to all other positions in the same sequence. The second sub-layer is a point-wise, fully connected FNN. Both sub-layers are followed by a residual connection

and layer normalization. The output of each layer is fed into the next layer, allowing the model to capture increasingly complex dependencies between the input tokens. The encoder is responsible for mapping an input sequence of symbol representations to a sequence of continuous representations, which are then used by the decoder to generate an output sequence of symbols.

2.6.1 Self-Attention

Self-attention is an attention mechanism that relates different positions of a single sequence in order to compute a representation of the sequence [2]. In the context of the Transformer network architecture, self-attention allows the model to weigh the significance of different words in a sentence when encoding or decoding the sequence. This mechanism computes the importance of each word in the sequence with respect to the other words in the same sequence, allowing the model to focus on different parts of the input when processing each word. This capability is crucial for capturing long-range dependencies and understanding the context of each word within the sequence. The attention mechanism is defined by the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, V and d are the query, key, value matrices and their dimensionality respectively. The softmax function is applied to the scaled dot-product of the query and key matrices, which produces a set of weights that indicate the importance of each key with respect to the query. The value matrix is then weighted by these weights and summed to produce the output of the attention mechanism.

2.6.2 Residual/Normalization

Residual connections, also known as skip connections, are used to address the vanishing gradient problem in deep neural networks [2]. They allow the output of a layer to bypass one or more layers and be added to the output of the deeper layer. This helps in preserving gradient flow during backpropagation and enables the network to learn more effectively, especially in the case of very deep networks.

Layer normalization is a technique used to normalize the inputs to a layer in a neural network. It operates on the principle of normalizing the activations of each layer across the feature dimension, which helps in stabilizing the training process and improving the

generalization performance of the model. By reducing the internal covariate shift, layer normalization can lead to faster convergence during training and better overall performance.

2.6.3 Linear layer

In the Transformer architecture, FFN is a component of each encoder and decoder layer [2]. Its role is to apply a non-linear transformation to the output of the self-attention sub-layer, which helps in capturing more complex relationships between the input and output sequences.

The FFN consists of two linear transformations with a ReLU activation function in between. Mathematically, the operation performed by the FFN can be expressed as:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

In this equation, x is the input to the FFN, W1 and W2 are the weight matrices of the two linear transformations, b1 and b2 are the bias vectors, and ReLU is the activation function.

3. Implementation

In this section will be discussed elements of transformer-decay model and how it works.

3.1 Datasets

For the purpose of training and evaluating our predictive models, we will utilize the MIMIC-III database, a comprehensive repository of anonymized health data sourced from critical care units at the Beth Israel Deaconess Medical Center between 2001 and 2012 [16]. This extensive dataset encompasses records from over forty thousand patients and includes a wide spectrum of clinical information, ranging from patient demographics to vital sign measurements, lab test results, medical procedures, prescriptions, healthcare provider observations, radiology reports, and mortality data, both within the hospital setting and following discharge.

It is worth mentioning that the predictions we perform are for patients in ICU. Indeed, MIMIC-III exclusively contains data on ICU patients. Predicting ICU readmissions is a specific case of hospital readmissions prediction. The significance of the MIMIC-III database lies in several key aspects: its accessibility to researchers worldwide, its representation of a diverse patient population admitted to intensive care units, and its meticulous documentation of patient-related data, facilitating comprehensive analyses encompassing vital statistics, laboratory findings, and medication records.

Table 1. Statistics of database.

Parameter	Numeric value
# of patients	14753
Gender distribution (female:male)	6691 : 8062
Average age at the time of visits	67
Average days per visit	10 days
# of unique medical codes	8993
# of unique ICD-9 diagnosis codes	6984
# of unique ICD-9 procedure codes	2009
# of unique prescription codes	2947

To make sure the relevance of our predictive modeling efforts, positive affected person cohorts could be excluded from consideration. Specifically, newborns and deceased sufferers will be excluded from the analysis, as readmission styles might not be relevant

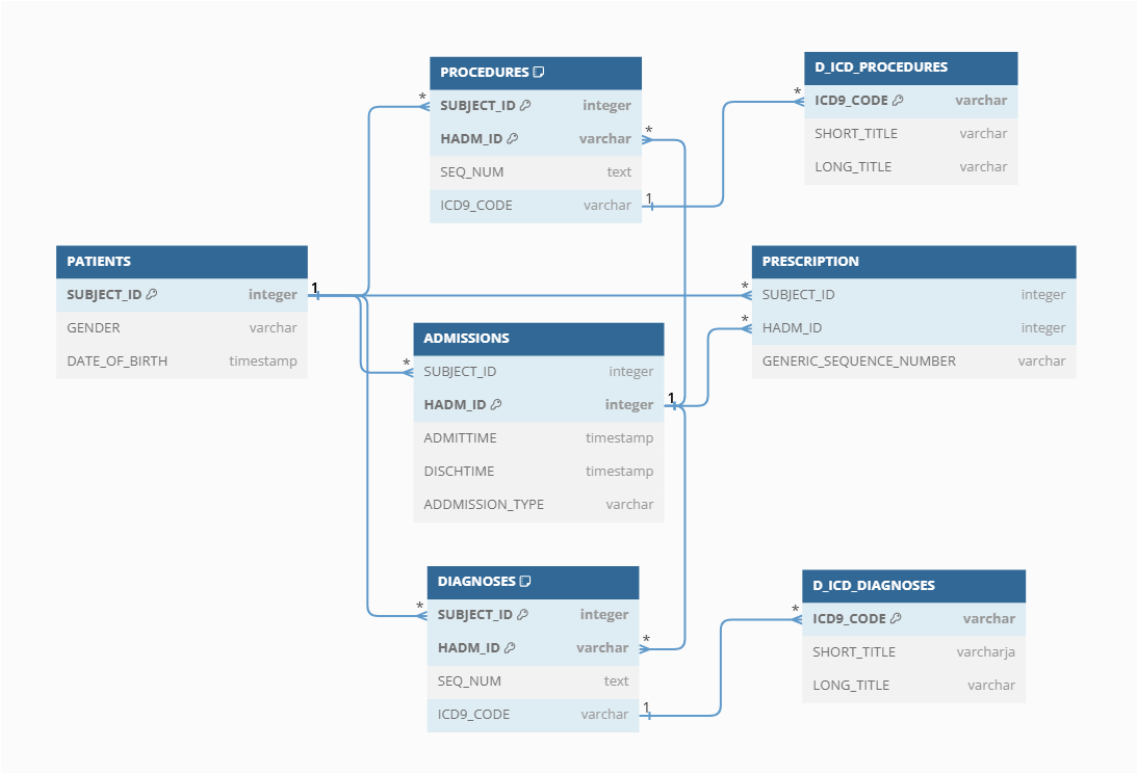

Figure 2. *Database architecture.*

in such cases. Additionally, we exclude patients under 18 years of age and restrict our analysis to patients with at least two admissions, as the predictive price of readmission likelihood will become greater significant in instances with repeated sanatorium encounters. Following this exclusion process, our dataset includes $14,753$ patients.

Additionally, we can restrict our analysis to patients with at least two admissions, as the predictive price of readmission likelihood will become greater significant in instances with repeated sanatorium encounters. Following this exclusion process, our dataset includes 14753 patients.

The structure of the dataset used for model training is depicted in Figure 2. Diagnoses and medical procedures, sharing a common structure, are amalgamated into a unified datatype for ease of analysis. Table 1 provides an overview of the statistical characteristics of the processed dataset, offering insights into its composition and distribution of key variables.

3.2 Environment

In this section will be discused the technologies were used during the research.

1. Python - programming language was used for scripting and algorithm implementation. It has become the de-facto language for training deep neural networks, coupling a large suite of scientific. computing libraries with efficient libraries for tensor computation [17].
2. PyCharm - IDE designed for Python programming with intuitive interface for code development, debugging, and organization.
3. Pandas and NumPy - Python's libraries used for data handling and manipulation [18, 19].
4. TensorFlow - a library for developing a machine learning applications, particularly excelling in deep learning and neural network implementations [20].

3.3 Setup configuration

For a better understanding of the experimental setup, we provide the details of the system configuration where the models were compiled and trained. The hardware and software specifications are important for reproducing the results and ensuring the model's performance consistency. The system configuration is shown in Table 2.

Table 2. System configuration.

Component	Specification
Processor	Intel Core i5-12400F
CPU Cores	6
RAM	32 GB
Operating System	Windows 10
Python	3.10
TensorFlow	2.15.0

3.4 Hyperparameters

This chapter outlines the chosen hyperparameters for our predictive models, focusing on their relevance and justification for optimizing model performance. The table 3 below summarizes the key hyperparameters used in our study.

We selected 100 epochs to allow sufficient training iterations. This balances underfitting, where the model might learn too little, and overfitting, where it might memorize noise in the data. This number of epochs ensures the model has ample opportunity to refine its weights and learn complex patterns.

Table 3. Model hyperparameters.

Parameter	N
Epochs	55
Batch patch	100
K-fold	5
Head attention	1
Optimizer	Adam
Learning rate	0.001
Transformer dropout rate	0.005

A single attention head was used in transformer-decay model to simplify the architecture. Preliminary experiments indicated that one attention head provides adequate performance without introducing unnecessary computational overhead, maintaining model interpretability.

Thirty linear units were included in the fully connected layers to capture complex relationships in the data while avoiding overfitting. This number was determined empirically and is supported by similar studies, providing a good trade-off between model capacity and complexity.

The selected hyperparameters are designed to optimize the model's performance efficiently. These choices balance complexity, computational efficiency, and the ability to generalize, aiming to build a robust model for predicting hospital readmissions.

3.5 Proof of Concept Models

In this chapter, we describe the proof of concept neural network model that will be applied to our dataset. The aim is to develop initial models that can establish a baseline for performance and provide insights into further improvements. It is important to note that the length of the input for one patient after the embedding will be 146, so the initial layer for every model will have 146 neurons.

3.5.1 Baselines

We compare our custom transformer-decay model with the following baseline models:

- **GRU-Decay**[21] is a GRU variant with an exponential decay function applied on the hidden layers;

- **LR** is a statistical method used for binary classification tasks, where the goal is to predict one of two possible outcomes. It models the probability of a given input belonging to a particular class by fitting data to a logistic function, which outputs values between 0 and 1.
- **Retain**[22] is an RNN-based model that integrates two attention mechanisms to capture the most relevant visits and clinical codes in the patient's longitudinal EHR;
- **Timeline**[23] is an RNN-based model that integrates an attention mechanism that calculates a new representation of a clinical code based on the visit context. Then, it calculates the effect of each diagnosis over time based on their initial influence value and their corresponding time decay factor;
- **LSTM** is an RNN-based model that uses memory cells to manage the flow of medical data, selectively retaining or discarding information as necessary.

3.6 Transformer

In this chapter, we describe the implementation of a Transformer-decay model. The following details outline the steps and considerations involved in the model development. Table 4 represents each layer of the model and its configuration. Every part of model has a 'get_config()' method to save trained model for following usage. We will separate and describe every significant step of the model.

Table 4. Transformer layers.

Layer	N
Embedding layer	8984 vocabulary size
Decay factor layer	80 dimensions
Transformer Block	80 dimensions
Demographic dense layer	15 neurons
Classifier	1 neuron

3.6.1 Embedding layer

We utilized TensorFlow's Embedding layer, specifying several key parameters to tailor its functionality to our needs. The input_dim parameter is set to the size of the vocabulary, which indicates the number of unique tokens the model can handle. This ensures that each unique token in our dataset is assigned a specific dense vector representation.

To handle sequences of different lengths effectively, we enabled the mask_zero parameter. By setting mask_zero=True, the embedding layer is instructed to ignore padding tokens, typically represented by zeros. This is particularly useful for managing input sequences of

varying lengths without introducing noise from the padding tokens.

Furthermore, we applied L2 regularization to the embedding weights using the embeddings_regularizer parameter. Regularization is a technique used to prevent overfitting by adding a penalty for large weights. In our implementation, L2 regularization is set with a factor of 0.02. This choice ensures that the model remains generalizable by discouraging excessively large weights, thereby promoting better performance on unseen data.

3.6.2 Decay factor layer

The layer is initialized with an output_dim parameter which specifies the dimensions of the output space. This dimension determines the shape of weights the decay factor will have. The layer is built with two matrices: weights and biases. The weight matrix, shaped (1, output_dim), is initialized using the Glorot uniform initializer. The bias vector, shaped (time_input.shape, 1), is initialized to zeros and adds a constant term to the computation.

When the layer is called, the time information is multiplied by the weight matrix and added to the bias vector. The result is passed through a ReLU activation function to ensure non-negative values. An exponential function is then applied to model the decay, exponentially decreasing the influence of admissions as they age, effectively capturing the essence of the time decay factor.

Then, result the decay factor layer is tiled to match the shape of the clinical code embeddings. The decay factor is concatenated with a tensor of ones that has the same shape as the decay factor. This step prepares the decay factor for integration with the clinical code embeddings. The ones ensure that the embeddings are not scaled down for any padding or zero entries in the clinical code sequence. Finally, the clinical code embeddings are element-wise multiplied by the decay factor. This step effectively adjusts the embeddings based on the time decay, reducing the influence of older admissions and thereby emphasizing more recent visits.

3.6.3 Transformer block.

The TransformerBlock class defines a custom layer for the Transformer model, incorporating multi-head attention, layer normalization, and dropout.

First, the MultiHeadAttention layer processes the input tensor twice (with query and key based on clinical code embeddings and decay factor vectors), generating an attention output.

This output represents the weighted sum of values, where the weights are determined by the query-key pairs. The attention output is then passed through the Dropout layer, which helps prevent overfitting by randomly setting a fraction of input units to zero during training. Finally, the combination of the initial input and the attention output is normalized using the LayerNormalization layer, stabilizing and speeding up the training process.

3.6.4 Integration

This section of the code integrates all the data and transforms it for the classifier. Initially, the model reduces the clinical code embeddings tensor along the first dimension (index 1), effectively summing the embeddings of clinical codes for each visit. If the clinical code embeddings originally have the shape (batch_size, num_visits, embedding_dim), after this operation, they will have the shape (batch_size, embedding_dim). This reduction is essential for condensing information from multiple visits into a single embedding vector per patient. A Dropout layer is then applied to these clinical code embeddings. The demographic data is processed through a dense layer, which likely transforms the demographic features. The resulting demographic tensor is concatenated with the clinical code embeddings tensor along the last dimension, merging the clinical and demographic information into a single tensor. Finally, this concatenated tensor is passed through an output layer with a sigmoid activation function, which outputs the probability of the target class (e.g., readmission risk).

3.7 Results and analysis

We dedicate this section to evaluating and comparing our model against the baselines listed in the previous section. We trained all models over 5-fold cross-validation and reported the average AUC and AUPRC scores in Table 5.

Table 5. AUC and AUPRC scores overs 5-fold cross validation of comptetiting models vs ours

Models	AUC	AUPRC
GRU-Decay	0.710 ± 0.011	0.345 ± 0.028
LR	0.664 ± 0.005	0.316 ± 0.016
Retain	0.682 ± 0.016	0.340 ± 0.013
Timeline	0.715 ± 0.007	0.355 ± 0.022
LSTM	0.699 ± 0.006	0.341 ± 0.014
Transformer-decay	0.699 ± 0.013	0.365 ± 0.020

$\pm$ Standard deviation.

Although our model does not achieve the highest AUC score, it outperforms all competing models in terms of AUPRC. This indicates its effectiveness in accurately identifying patients who will be readmitted. The AUC score measures the model's ability to correctly classify the negative class (patients who will not be readmitted), while the AUPRC reflects the model's capability to correctly classify the positive class (patients who will be readmitted). Given the context, the AUPRC score is arguably the more critical metric because we aim to minimize the rate of false negatives (patients who should be readmitted but are misclassified as not needing readmission), as these errors could lead to severe consequences, such as death.

For practical applications, evaluating the scalability of these models in real-world scenarios is crucial. This includes assessing their performance on larger datasets, their ability to integrate with existing healthcare systems, and the feasibility of deploying them in a production environment.

It is important to note that the database used for this study turns out to be small for such a complicated task. Given the scale, the results are satisfying but it is necessary to repeat the research with a larger and more recent database to ensure the robustness and generalizability of the findings.

3.8 Advantages and limitations

The application of transformer-decay models for readmission prediction offers several significant advantages. This model addresses numerous issues within hospital settings. By predicting readmissions accurately, it helps in the proactive management of patient care, potentially reducing the number of unexpected readmissions. By reducing the rate of readmissions, hospitals can lower healthcare costs for both patients and institutions. Better prediction of readmissions allows for more effective management of hospital resources, ensuring that beds, medical staff, and equipment are utilized optimally. Accurate readmission predictions can lead to timely interventions, reducing patient mortality rates and enhancing the overall quality of life for patients by ensuring they receive the necessary care when needed. By preventing unnecessary readmissions, the model helps in decongesting hospitals, which is particularly beneficial during peak times or pandemics when hospital resources are stretched thin. The transformer-decay model excels in handling complex text data, such as medical procedures and diagnoses, making it particularly suitable for medical applications where patient records and clinical notes are pivotal. The inclusion of a time decay factor ensures that the model emphasizes the most recent admissions, which are more reflective of current medical practices and treatments. This approach helps maintain the relevance and accuracy of predictions.

Despite these advantages, there are several limitations to the current work. The model did not achieve the best results in terms of AUC compared to some other models. This indicates a need for further refinement and improvement in its ability to distinguish between patients who will and will not be readmitted. The training data used for this study was relatively small for such a complex task. A larger and more recent dataset is necessary to improve the robustness and generalizability of the findings. This limitation highlights the need for additional data collection and validation on more extensive datasets. Given the small size of the training dataset, there is a risk of overfitting, where the model learns the specifics of the training data too well but fails to generalize to new, unseen data. The transformer-decay model, while effective, requires substantial computational resources and longer training times. This can be an obstacle to its practical implementation in resource-limited settings. The complexity of the transformer model can make it difficult to interpret and explain its predictions. Enhancing the interpretability of the model is crucial for gaining insights into the factors contributing to readmissions and for fostering trust among healthcare professionals. The performance of the model might vary across different hospital settings and patient populations. It is important to validate the model across diverse clinical environments to ensure its widespread applicability and effectiveness.

4. Summary

The primary objective of this thesis was to implement the transformer-decay model and compare its performance with classical machine learning architectures. To achieve this, the database was parsed and preprocessed to ensure it was suitable for model training and comprehension. All implementation was carried out in Python, utilizing TensorFlow as the main framework for developing neural network models.

We started by implementing simple proof-of-concept models based on related works. The transformer model, incorporating a time decay layer, was successfully implemented. The data was divided into 5 folds for training and testing to ensure robust evaluation.

Appropriate metrics were selected to assess the performance of the models. The transformer-decay model demonstrated a high result in AUPRC, indicating its effectiveness in handling imbalanced datasets, such as those encountered in medical prediction where positive cases are rare. However, its performance on AUC was mediocre, suggesting room for improvement in distinguishing between classes.

In conclusion, the goal of this work has been achieved. The implemented transformer-decay model shows significant potential for future development. Future researchers are encouraged to utilize larger and more recent databases to further enhance the model's accuracy and applicability.

References

[1] Clare Allison Parker et al. “Predicting hospital admission at the emergency department triage: A novel prediction model”. In: *The American Journal of Emergency Medicine* 37.8 (2019), pp. 1498–1504.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[3] Xu Min, Bin Yu, and Fei Wang. “Predictive Modeling of the Hospital Readmission Risk from Patients’ Claims Data Using Machine Learning: A Case Study on COPD”. In: *Scientific Reports* 9.1 (Feb. 2019).

[4] M. Pishgar et al. “Prediction of unplanned 30-day readmission for ICU patients with heart failure”. In: *BMC Medical Informatics and Decision Making* 22.1 (May 2022).

[5] Awais Ashfaq et al. “Readmission prediction using deep learning on electronic health records”. In: *Journal of Biomedical Informatics* 97 (2019), p. 103256.

[6] Yuhan Deng et al. “Explainable time-series deep learning models for the prediction of mortality, prolonged length of stay and 30-day readmission in intensive care patients”. In: *Frontiers in Medicine* 9 (Sept. 2022).

[7] Tian Bai et al. “Interpretable Representation Learning for Healthcare via Capturing Disease Progression through Time”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018.

[8] Cao Xiao et al. “Readmission prediction via deep contextual embedding of clinical concepts”. In: *PLOS ONE* 13.4 (2018), e0195024.

[9] Ahmad Hammoudeh et al. “Predicting Hospital Readmission among Diabetics using Deep Learning”. In: *Procedia Computer Science* 141 (2018), pp. 484–489.

[10] Ricardo M. S. Carvalho, Daniela Oliveira, and Catia Pesquita. “Knowledge Graph Embeddings for ICU readmission prediction”. In: *BMC Medical Informatics and Decision Making* 23.1 (Jan. 2023).

[11] Chien-Yu Chi et al. “Predicting the Mortality and Readmission of In-Hospital Cardiac Arrest Patients With Electronic Health Records: A Machine Learning Approach”. In: *Journal of Medical Internet Research* 23.9 (2021), e27798.

[12] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. “ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission”. In: (2019).

[13] Chuhong Lahlou et al. “Explainable Health Risk Predictor with Transformer-based Medicare Claim Encoder”. In: (2021).

[14] Yan Miao and Lequan Yu. “MuST: Multimodal Spatiotemporal Graph-Transformer for Hospital Readmission Prediction”. In: (2023). URL: http://arxiv.org/pdf/2311.07608.

[15] Weimin Lyu et al. “A Multimodal Transformer: Fusing Clinical Notes with Structured EHR Data for Interpretable In-Hospital Mortality Prediction”. In: *AMIA Annual Symposium Proceedings* 2022 ().

[16] *MIMIC-III Clinical Database v1.4*. https://physionet.org/content/mimiciii/1.4/. 2016.

[17] Zachary DeVito, Jason Ansel, Will Constable, et al. “Using Python for Model Inference in Deep Learning”. In: (2021).

[18] Aritra Pain. *Pandas Library in Data Science: Harnessing the Power of Data Manipulation*. https://www.linkedin.com/pulse/pandas-library-data-science-harnessing-power-aritra-pain/. 2023.

[19] Stéfan van der Walt, S Chris Colbert, and Gaël Varoquaux. “The NumPy Array: A Structure for Efficient Numerical Computation”. In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[20] Nidhin Mahesh. *Understanding a TensorFlow program in simple steps. | Towards Data Science*. https://towardsdatascience.com/understanding-fundamentals-of-tensorflow-program-and-why-it-is-necessary-94cf5b60e255. 2017.

[21] Zhengping Che et al. “Recurrent Neural Networks for Multivariate Time Series with Missing Values”. In: *Scientific Reports* 8.1 (Apr. 2018).

[22] Edward Choi et al. “Retain: An interpretable predictive model for healthcare using reverse time attention mechanism”. In: *Advances in neural information processing systems* 29 (2016).

[23] Tian Bai et al. “Interpretable Representation Learning for Healthcare via Capturing Disease Progression through Time”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*. ACM, July 2018.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Pavel Grubeljas

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Transformer-based model for predicting hospital readmissions", supervised by Sadok Ben Yahia and Nzamba Bignoumba
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

27.05.2024

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Certificate

Figure 3. *Certificate of completing Data or Specimens Only Research Course.*