Suicide Risk Prediction using Electronic Health Records

Master Thesis

Author(s):
Ancillon, Lou

Publication date:
2024

Permanent link:
https://doi.org/https://doi.org/10.3929/ethz-b-000673932

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

HARVARD MEDICAL SCHOOL
BLAVATNIK INSTITUTE
BIOMEDICAL INFORMATICS

Suicide Risk Prediction using Electronic Health Records

Master Thesis

Lou Ancillon

April 2024

Supervisors: Prof. Joachim M. Buhmann, Prof. Tianxi Cai
Advisors: Robin Geyer, Clara-Lea Bonzel

Department of Computer Science, ETH Zürich

Abstract

Background: Suicide is a growing public health concern. Accurately predicting suicide risk can enable clinicians to diagnose and intervene early. Since nearly half of suicide victims visit their physician within one month before attempting suicide, electronic health records (EHRs) serve as a valuable resource for predicting suicide risk. This study aimed to compare the performance of traditional and advanced machine learning models in predicting suicide risk using EHR data.
Methods: We re-implemented and extended a regularized Cox model to include a wider range of EHR data types, such as diagnosis, medication information, procedure, laboratory tests, and clinical notes. In addition, we adopted a transformer approach, BRLTM, based on the BEHRT architecture, and developed a novel model, Multiseq, to address the challenges posed by EHR data, such as complex relationships between past, present, and future codes, irregular intervals between visits, and concept codes of different sizes.
Results: The results showed that using multiple types of codes in the Cox model lead to better performances, demonstrating the link between physical and mental health in predicting suicide. Transformer models outperformed the regularized Cox regression model due to their ability to handle the temporal sequence of EHR data and capture complex interactions within patient records. Feature selection significantly improved the performance of both transformer architectures.
Conclusion: This study represents a significant advance in the use of EHRs for suicide risk prediction. We demonstrated the improved performance of transformer models in capturing the complex nature of EHR data compared to traditional models. We also demonstrated the importance of integrating multiple types of EHR data to improve clinical decision making for SRB diagnosis. Limitations and future work were discussed, including the need for external validation, improved interpretability, and exploration of alternative feature selection strategies.
Keywords: Suicide Risk Prediction ; Electronic Health Records (EHR) ; Machine Learning ; Regularized Cox Model ; Transformer Models ; BEHRT ; Multimodal Data Integration ; Feature Selection ; Mental Health Care.

Contents

Chapter 1

Introduction

1.1 Presentation of the project

Between 2002 and 2021, suicide rates experienced a 36% surge (see figure 1.1), impacting over 700,000 individuals annually by 2021 [1]. Moreover, 2021 data shows that 12.3 million American adults had serious suicidal thoughts, with 3.5 million making plans and 1.7 million attempting suicide [2]. These alarming numbers underscore the need to address not only suicide deaths, but also all forms of suicide-related behavior (SRB).

Meanwhile, using data from electronic health records (EHRs) has become an important resource for large, real-world population studies, providing cost-effective alternatives to traditional research methods [3]. As shown in Figure 1.2, EHR adoption in the US grew exponentially between 2008 and 2014, in both hospitals and physician practices. Since 2014, this growth has leveled off, with over 95% of hospitals and around 75-80% of office-based physicians using EHRs.

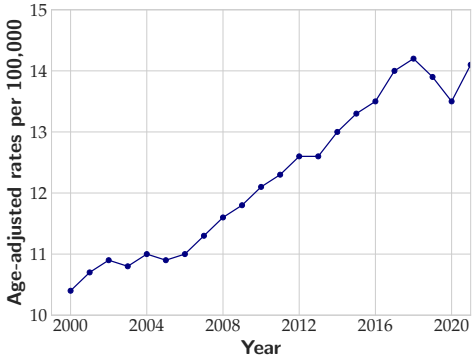


Figure 1.1: Evolution of the number of suicides in the US [2]

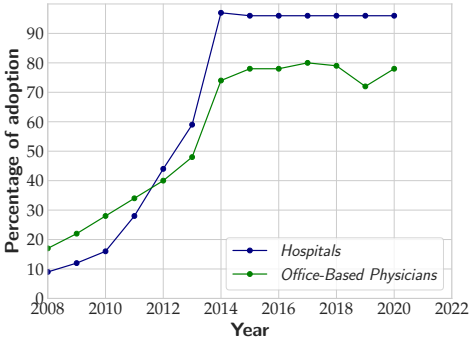


Figure 1.2: Evolution of the use of EHR in the US [4]

This study explores predictive models for SRB to enhance clinicians' preventive strategies. The approach used is based on the landmark model proposed by Sheu et al. [5], which ensures that the distributions of patient training data are comparable between those who experience an SRB event and those who do not. This is achieved by randomly selecting a date in the patient sequence and making predictions from that point forward, without differentiating between cases and controls in the selection of data. This systematic approach is versatile and can be adapted to various statistical or machine learning (ML) model architectures, given that they are appropriate for the specific task and dataset at hand.

Our work will first extend the cox regression model from Sheu et al. [5] by incorporating additional types of patient data, such as medical procedures, prescribed medications, laboratory test, and Concept Unique Identifiers (CUIs) extracted from medical notes. By integrating these diverse data sources, we hope to gain a more comprehensive understanding of patient outcomes and improve the accuracy of our predictions. We will also evaluate the impact of existing feature selection methods [6, 7]. However, this initial model has limitations in accounting for the chronology of events and capturing complex long-term dependencies between clinical codes.

To overcome these limitations, we will investigate the use of transformers, specifically Bidirectional Encoder Representations from Transformers (BERT) models, for EHR data, adapting landmark model methodology to better understand the interrelations of codes and their temporal effects on patient outcomes.

1.2 Definitions

To provide context for our work, we begin by defining suicide and suicide-related behaviors (SRB), as these terms may not be familiar to everyone. We also describe Electronic Health Records (EHRs) and the types of data they encompass. This information is crucial for understanding the scope of our study and the data we use to analyze SRB.

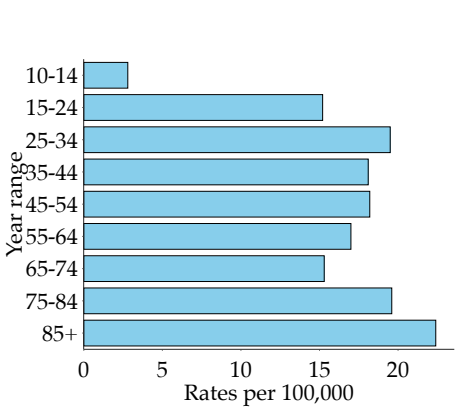
1.2.1 Suicide related behaviors

A suicide attempt is generally defined as an act of self-harm with varying degrees of intent to end one's life. Suicidal behavior includes both the act of suicide itself and a spectrum of attempts, from those that are highly dangerous to less lethal acts that often occur during psychosocial crises and sometimes signal a request for help. In the context of suicide-related behaviors, we also consider suicidal ideation which includes thoughts, wishes, and preoccupations about suicide. This forms a continuum that includes

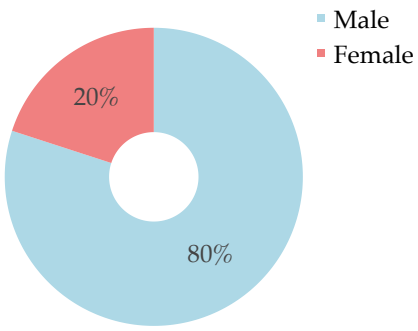
ideation, planning, non-lethal attempts, and completed suicide, although the progression is not always sequential [8, 9, 10].

Suicide is frequently linked to psychiatric disorders, with over 90% of individuals who die by suicide having a diagnosed mental health condition, a pattern that often holds true for those who attempt suicide as well. Mood disorders represent the most prevalent category of conditions associated with suicide and serious suicide attempts, accounting for 60% of all cases. Additionally, personality disorders, substance abuse, anxiety disorders, and schizophrenia are all significant contributing factors to suicidal behavior [11, 10].

The figures 1.3a and 1.3b show some statistics about how the population is affected by suicide. Globally, suicide rates are higher among men than women, but women are more likely to attempt suicide. The highest suicide rates are found in individuals aged 70 and above. However, there is a growing concern over rising suicide rates among the young, with suicide being the second leading cause of death among 15 to 29-year-olds [2].



(a) Rates of suicides according to the age [2]



(b) Proportion of suicides according to the gender [2]

Figure 1.3: Suicide population statistics

1.2.2 Electronic Health Record

EHR data elements include both structured and unstructured data. Structured elements include demographic data such as age, biological sex, race/ethnicity, language, insurance type, and more. It also represents clinical information such as medication prescriptions, lab results, and ICD-10 codes. On

the other hand, unstructured elements include images or clinician notes.

The real-world data from EHRs is inherently complex and typically requires careful curation and processing to transform the raw characteristics present in the data into relevant concepts and variables of interest. To develop scalable and generalizable ML algorithms, we will employ the codes resulting from an ontology-driven approach. This approach transforms source codes from the EHR into standardized groupings such as Phecodes mapped from ICD codes, according to PheWAS Resources [12], Ingredient-level RxNorm codes mapped from medication prescriptions [13], Grouped CPT procedure codes according to Clinical Classifications Software [14], and LOINC codes mapped from laboratory tests [15]. These groupings are based on vocabulary mappings and are regularly updated to incorporate new codes. This optimization facilitates effective feature selection and the development of meaningful feature space embeddings.

1.3 Introduction and Motivation

1.3.1 Background

Suicide is a major public health crisis, currently ranked as the 18th leading cause of death worldwide and the second leading cause of death among persons aged 15 to 29 years [1]. The role of primary care physicians is paramount, as they see 45% of individuals who die by suicide within one month prior to their act [16]. Health care settings are therefore a key venue for identifying and intervening to prevent SRBs.

Educating and training primary care providers to better recognize and treat depression has been highlighted as a critical method for reducing suicide rates. However, even with these educational initiatives, clinicians' ability to accurately predict suicide risk remains a challenge, often comparable to chance. Physicians have the critical role of collecting and analyzing patients' clinical features, documenting these observations, and using this information to formulate decisions about the patient's risk to self and the appropriate treatment plan [17, 18].

It has been shown by [19] that suicide risk assessments should at least consider several key factors. These include previous suicide attempts, current symptoms of mood and anxiety disorders, coexisting psychiatric conditions such as substance abuse disorders, the effects of medications, the absence of social or family support networks, and the patient's access to potentially lethal means. It is also important to assess whether the patient has formulated a suicide plan.

According to [10], the most reliable indicators of suicide are a history of previous attempts or the presence of suicidal thoughts. In fact, specific studies [16, 18] have identified the three most significant predictors of future suicidal acts among individuals suffering from major depressive or bipolar disorders. These include a history of previous suicide attempts, subjective ratings of depression severity, and cigarette smoking, all of which cumulatively increase the risk of future suicidal behavior.

In practice, clinicians depend on and rely on information voluntarily provided by patients about past personal suicidal behavior or a family history of suicidal behavior. Unfortunately, some patients, especially those most determined to commit suicide, may deny their true feelings of suicidal ideation out of concern that their plans will be prevented [10]. As a result, many individuals at risk for suicide remain undiagnosed or untreated for their mental health conditions.

With EHRs, we now have more patient and clinical data than ever before. EHRs help keep detailed records of a patient's medical history and collect important health information. We can use this data to build strong ML models that can help find diseases, decide on the right treatment, and most importantly, figure out if a patient might be at risk of suicide, even if they are in denial. The use of ML models with large clinical datasets has shown promising results [20, 21, 22], often outperforming traditional clinician performance in predicting the risk of developing certain conditions. This new method could change the way physicians identify and help patients who may be experiencing SRB, providing an important way to identify and treat mental health conditions early.

1.3.2 Objective

The goal of this work is to improve the performance of innovative predictive modeling strategies for suicide risk. This will be achieved by developing and evaluating novel statistical and ML approaches. Specifically, an interpretable ML model, the regularized Cox model, will be reimplemented and extended to build predictive models for suicide risk. This interpretable ML model will be compared to more complex ML models such as transformers, which can capture the challenges posed by EHR data, such as complex interactions between past, present, and future codes, long-term dependencies between codes, and irregularities between visits. Transformers allow the chronological order of visits to be maintained, providing a more accurate representation of a patient's condition over time.

The final models will be compared on the basis of their interpretability, insight, and predictive performance. The goal is to implement models that can

improve current suicide risk prediction and support clinician decision making. The models should provide insights into which features are relevant for suicide risk detection, how risk evolves over the course of a patient's life, and how to prevent a suicide event. By replicating real-world scenarios faced by clinicians, this work aims to make a valuable contribution to the field of suicide risk prediction and prevention.

Chapter 2

Literature Review

The increasing prevalence of mental disorders, suicides, and SRBs, coupled with the difficulty of diagnosing these conditions [17, 18], has led to an increase in ML applied to psychiatric studies in recent years. In fact, a systematic review by Piggoni et al. [23] found that half of the referenced reports were published between 2019 and 2022. Rather than considering both healthy and psychiatric patients, which could hinder the identification of subtle risk factors specific to various psychiatric disorders, the review focuses solely on vulnerable populations.

ML has the potential to identify predictors of suicidal behavior specific to certain disorders and to help stratify patients according to actual risk [24]. This review [23] analyzes over a thousand studies and highlights the potential of ML models such as Random Forest, Support Vector Machine, and Convolutional Neural Networks to outperform traditional clinical assessments for suicide risk stratification. The review allows comparison of a large number of studies on several aspects, including the outcome used, the ML algorithms employed, the number and type of features, and the sample size. The majority of studies predict lifetime suicide attempts using, for the majority, clinical and sociodemographic variables found in EHR data [25].

In the same direction as the work presented in the systematic review, Sheu et al. [5] implemented a model that differs from conventional ML models, which may suffer from temporal bias due to case-control sampling. This landmark model results from an approach that treats cases and controls equally. The framework randomly selects a time point for each patient and makes predictions over specified prediction windows (ranging from six months to 1.5 years) using historical data (ranging from 3 months to the patient's entire medical history). This setting is more in line with clinical practice, where clinicians want to determine whether a patient is at risk of attempting suicide within a given time window without having access to the

patient's case or control status. The author uses only demographic information and diagnosis codes (PheCode) [12], not including additional types of information such as medications, procedures, lab, and CUIs. which might be a limitation of this approach.
For each patient, the total number of each PheCode during the look-back period is extracted, normalized to the available observation period length so that it is not biased by the different length of medical history between patients, and then log-transformed.
The application of regularized Cox regression and random survival forest within this framework shows promising performance across different patient cohorts and prediction windows, establishing a fundamental benchmark for our investigation.

While the landmark model provides a robust baseline, it has limitations because it does not account for the temporal evolution of a patient's state. To gain a more complete understanding of the complex progression of patient states and to gain deeper insight into patient data, we explored models that take this into account.
The advent of transformer-based models provides an opportunity to refine and improve predictive analytics in healthcare. In our research, we aimed to investigate models capable of capturing the complex temporal evolution of patient states, ultimately providing deeper insights into patient data [26, 27, 28, 29, 30].

With the introduction of BEHRT [26], the authors propose a novel NLP-inspired way to represent EHR patient sequence: diagnoses as words, each visit as a sentence, and a patient's entire medical history as a document.
Each patient's EHR is represented as $V_p = \{v_p^1, v_p^2, ..., v_p^{n_p}\}$, where n_p is the number of visits in patient p's EHR, and v_p^j contains the diagnoses in the jth visit, which can be thought of as a list of m_j diagnoses.
To prepare the data for BEHRT, we order the visits (and thus the diagnosis codes) in time, and introduce a term CLS to denote the start of the medical history, and the space between visits SEP, resulting in a new sequence, the patient's EHR sequence $V = \{CLS, v_1, SEP, v_2, SEP, ..., v_{np}, SEP\}$.

Multiple embeddings are then used to represent EHR data:

- Code Embedding: similar to word embeddings in NLP.
- Age Embedding: Embedding age and linking it to each visit serves two purposes: providing the model with a sense of time, and informing it of age as a health notion of when things happened. Additionally, age is an important factor in health care, influencing the risk of various diseases.

- Position Embedding: Determines the relative position of concepts in the EHR sequence and enables the network to capture the positional interactions between diseases. Within a single visit, there is no predefined order for codes; therefore, position embedding primarily captures the temporal progression of visits rather than the arrangement of codes within a visit (i.e., position embedding corresponds to the visit index in a patient's timeline).
- Segment Embedding: alternates between visits and is used to provide BEHRT with additional information to indicate the separation between visits.

The combination of these embeddings provides a rich, multidimensional representation of EHR data that captures both clinical concepts and their temporal context. The model is then pre-trained using a Masked Language Model (MLM) task, which helps the model learn rich representations of diagnosis and patient states that capture underlying patterns in the EHR data. After MLM pre-training, BEHRT can be considered a versatile model that can be applied to a variety of downstream tasks with minimal additional training. In the original paper, the model is fine-tuned to predict the likelihood of 301 conditions in future patient visits. However, other studies, such as [31], have used a similar architecture and pre-training approach to BEHRT, but have fine-tuned the model for alternative purposes, such as detecting the onset of depression.

BEHRT can be seen as a good baseline for further improvements. The BERT [32] architecture allows to capture the complex and non-linear interactions between past, present and future concepts, as well as the long-term dependencies between concepts. It is still limited to a single code type and a small vocabulary, but has a flexible architecture. It also remains constrained in its ability to capture temporal information, as it does not account for the irregular intervals between consecutive patient visits.

To overcome these limitations, CEHR-BERT [30] aims to improve disease prediction by modifying the patient sequence representation and incorporating temporal information more effectively through age and time embeddings. The sequence representation is similar to BEHRT, but with all concepts encoded together and additional emphasis on encoding temporal information. Instead of a SEP token, artificial time tokens (ATT) are inserted between visits to denote time intervals. The patient sequence representation is then noted as $P = \{VS, v_1, VE, ATT, VS, v_2, VE, ATT, ..., VS, v_i, VE\}$. Where VS, and VE stand for Visit Start and Visit End respectively.
They encode both absolute time (time embeddings) and relative time (age) with respect to the visit embedding, which they concatenate with the con-

cept embedding. This embedding passes through a fully connected layer that adjusts the dimensionality back to the original size, creating a temporal concept embedding. The FC layer can learn interactions between different embeddings (e.g., between time embeddings and concept embeddings), as opposed to summing the embeddings.
As a primary learning objective, CEHR-BERT uses MLM to pre-train the model. As a novel secondary learning objective, Visit Type Prediction is introduced to improve performance in downstream tasks by leveraging domain knowledge about the association between medical concepts and visit types.
By incorporating temporal information and adapting the BERT architecture to better fit structured EHR data, CEHR-BERT demonstrates enhanced predictive capabilities, proving its potential to provide more accurate and timely predictions for various medical conditions and patient outcomes. It addresses the challenge of irregular visits by incorporating *ATT* and providing more temporal information. However, adding new tokens to the patient sequence significantly increases the length of the sequence and may result in loss of information since the maximum length accepted by BERT is 512 tokens.

This limitation is solved by the architecture proposed in ExBERT [27], an extension of BEHRT where medical concepts are not concatenated into one long vector, but grouped into separate, learnable embeddings per concept type. This avoids exploding input lengths when adding new medical features, and allows the model to learn which concepts to focus on. From a clinical perspective, it makes sense to separate diagnoses, procedures, drugs, etc., as they have different clinical value for downstream applications.

Deep learning, when combined with survival analysis, can significantly improve predictive accuracy [33, 34]. It can also offer better insights of how the relevance and significance of different features change over time in predicting the outcome [27]. Kvamme et al. [35] introduce methodologies that employ a scalable loss function, derived from nested case-control studies, allowing for both proportional and non-proportional extensions of the Cox model. The BERTSurv framework [36] takes traditional deep learning survival models a step further by integrating unstructured clinical notes with structured EHR data. This approach enhances both mortality prediction and survival analysis by making use of binary cross-entropy loss for mortality prediction and partial log-likelihood loss for survival analysis.

Our research contributions are twofold. First, we aim to improve upon the existing landmark model by incorporating additional types of patient data and exploring feature selection strategies [6, 7]. This enhanced version of the

landmark model will serve as a new state-of-the-art baseline for predicting SRBs.

Our main contribution is a novel transformer-based architecture, inspired by BEHRT [26], that creates a new representation of EHR sequences suitable for our task of predicting SRBs in different time windows. This will be achieved by combining the different approaches described earlier in the literature [31, 30, 27]. To the best of our knowledge, ours is the first work to apply transformers to SRB prediction, demonstrating significantly improved performance for a variety of prediction windows.

Chapter 3

Method

3.1 Dataset

The datasets for this study were obtained from the Mass General Brigham (MGB) Biobank [37] and the Research Patient Data Registry (RPDR) [38]. The MGB Biobank is a large-scale research program focused on understanding the influence of an individual's genes, lifestyle, and environment on health. It stores health information on more than 145,000 patients and supports more than 450 studies. On the other hand, the RPDR is a centralized data repository that collects clinical information from the EHRs of the MGB system. This database contains data from more than seven million patients, with more than three billion medical records from eight affiliated hospitals, including two major teaching hospitals: Massachusetts General Hospital (MGH) and Brigham and Women's Hospital (BWH).

The Biobank dataset was used for the SRB risk prediction task in both the regularized Cox regression model and the fine-tuning of the transformer-based models. On the other hand, the RPDR dataset was used exclusively for the pre-training of the transformer models.

3.1.1 Features used

Our study leveraged diverse types of patient information, critical for training our predictive models:

- Demographic information: Includes age, gender, race, and marital status for each patient.
  Understanding demographic factors is important because they can significantly influence mental health and suicide risk. For example, certain age groups or racial backgrounds may be more vulnerable to suicide risk due to sociocultural, environmental, or genetic factors. In

addition, gender and marital status can also influence an individual's mental well-being and susceptibility to suicidal behavior.

- Diagnosis Codes (PheCodes) [12] : These codes are used to classify patient diagnoses into broader categories for research, derived from ICD codes.
  PheCodes help identify potential phenotypic correlations with suicide risk by grouping related medical conditions.
- Medication Codes (RxNorm) [13] : RxNorm provides a standardized nomenclature for medications, enabling clear communication and consistency across healthcare settings.
  Understanding medication history is critical, as certain medications may have side effects that affect mental health, or may indicate treatment for existing mental health conditions.
- Procedure Codes (CPT/CCS) [14] : CPT (Current Procedural Terminology) and CCS (Clinical Classifications Software) codes represent specific medical procedures received by patients.
  Analyzing these codes helps identify treatments or surgeries that may be related to or impact SRB risk.
- Laboratory Codes (LOINC) [15] : LOINC codes standardize terms for laboratory tests and observations.
  This information can reveal underlying medical conditions or the effects of medications.
- Clinical Codes from Medical Notes (CUI): Concept Unique Identifiers (CUI) from the Unified Medical Language System (UMLS) are used to encode clinical concepts found in medical notes.
  These codes facilitate the analysis of detailed patient narratives, providing deeper insight into patient history, symptoms, and other clinical events relevant to making accurate predictions.

These diverse datasets and feature types are essential for developing robust suicide risk prediction models. They provide a comprehensive view of each patient's clinical profile, enabling more accurate and personalized risk assessments.

3.1.2 SRB outcome

In our study, cases were defined as patients who had at least one SRB event in a specific predictive window. That is, having at least one of these codes in their EHR sequence:

- PheCode 297 : Suicidal ideation or attempt
- PheCode 297.1 : Suicidal ideation

- PheCode 297.2 : Suicide or self-inflicted injury

Broader category codes such as PheCodes can be useful in identifying trends, patterns, and associations that may not be apparent at the individual ICD code level. However, these categories provide a more general perspective and may lack the specificity and detail required for certain analyses. In fact, in the context of SRB, some PheCodes are overly broad, causing us to miss certain events. For example, PheCode 965.3 includes the ICD codes poisoning by salicylates (965.1), accidental poisoning by salicylates (E850.3), and salicylates causing adverse effects in therapeutic use (E935.3). While these three ICD codes may have the same effect in some contexts, for our study the first one should be classified as suicide, unlike the other two.

3.1.3 Data Sample Description

Before delving into the implementation, it is important to have some context about the datasets used in the study. This section presents some statistics about the BioBank and RPDR datasets.

BioBank Dataset

The initial dataset consisted of 125,395 patients. Following the principle mentioned by Piggoni et al. [23], we first filtered the patients to retain only those with at least one Mental Health (MH) PheCode (see table 3.1). This resulted in a dataset of 60,820 patients. The idea behind this is that when large sample studies of MH and non-MH populations are combined, a formal MH diagnosis becomes the main predictor of suicide risk, eclipsing more nuanced risk factors specific to different populations. From these MH patients, we further selected those who had at least 3 visits and 3 months of data history to ensure that the model could learn from each patient to make predictions. This resulted in a final dataset of 59,411 patients.

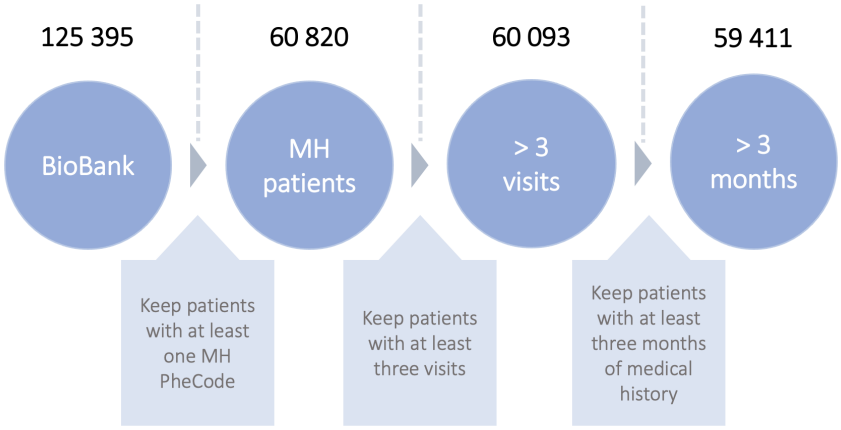


Figure 3.1: Filtering of BioBank data. This flow lists all the key steps of our data cleaning. At each step, the number of patients that are included is shown. As you can see, we started with nearly 125 thousands patients and our final data (used for training and evaluation of our models) consists of 59 thousands patients, each meeting our inclusion criteria.

PheCode	Description	PheCode	Description
295	Schizophrenia and other psychotic disorders	300.4	Dysthymic disorder
295.1	Schizophrenia	300.8	Acute reaction to stress
295.2	Paranoid disorders	300.9	Posttraumatic stress disorder
295.3	Psychosis	301	Personality disorders
296	Mood disorders	301.1	Schizoid personality disorder
296.1	Bipolar	301.2	Antisocial/borderline personality disorder
296.2	Depression	302	Sexual and gender identity disorder
296.22	Major depressive disorder	302.1	Decreased libido
297	Suicidal ideation or attempt	303	Psychogenic and somatoform disorder
297.1	Suicidal ideation	303.1	Dissociative disorder
297.2	Suicide or self-inflicted injury	303.3	Psychogenic disorder
300	Anxiety, phobic, and dissociative disorders	303.31	Gastrointestinal malfunction arising from mental factors
300.1	Anxiety disorder	303.4	Somatoform disorder
300.11	Generalized anxiety disorder	304	Adjustment reaction
300.12	Agoraphobia, social phobia, and panic disorder	305.2	Eating disorder
300.13	Phobia	305.21	Anorexia nervosa
300.2	Generalized anxiety and phobic disorders	300.3	Obsessive-compulsive disorders

Table 3.1: MH PheCodes and their descriptions

Table 3.2 presents some statistics about the population. The dataset predominantly consist of females (61.3%) compared to males. The majority of patients are either married or single, and most are white.

Setting	Number	Percentage
Gender		
Female	36,421	61.3%
Male	22,988	38.7%
Race		
Asian	1,181	2.0%
Black	3,220	5.4%
White	50,431	85.0%
Other	2,293	3.9%
Unknown	2,286	3.8%
Marital Status		
Widow	2,790	4.7%
Married	29,771	50.1%
Divorced	5,073	8.5%
Single	19,537	32.9%
Other	19,491	32.8%
Age		
18-35	8,349	14.0%
35-55	16,268	27.4%
55-70	19,163	32.3%
70+	15,631	26.3%

Table 3.2: Biobank Population Statistics

In terms of age, most patients are between 35 and 70 years old, with the highest SRB prevalence in the 35-55 age group and the lowest in the 70+ age group. The overall prevalence of SRB in the data set was 18.9%.

Age category	Number	Percentage
All	11,227	18.9%
18-35	2,019	24.2%
35-55	2,999	18.4%
55-70	3,231	16.9%
70+	2,978	19.1%

Table 3.3: SRB prevalence per age

The bar chart 3.2 shows the number and proportion of patients who experienced each of the different MH PheCodes. Most patients experienced anxiety and phobia (300, 300.1), mood disorders (296, 296.2, 296.22), and adjustment reactions (304). This is noteworthy because mood disorders are known to be one of the primary pathologies associated with SRB in the literature.

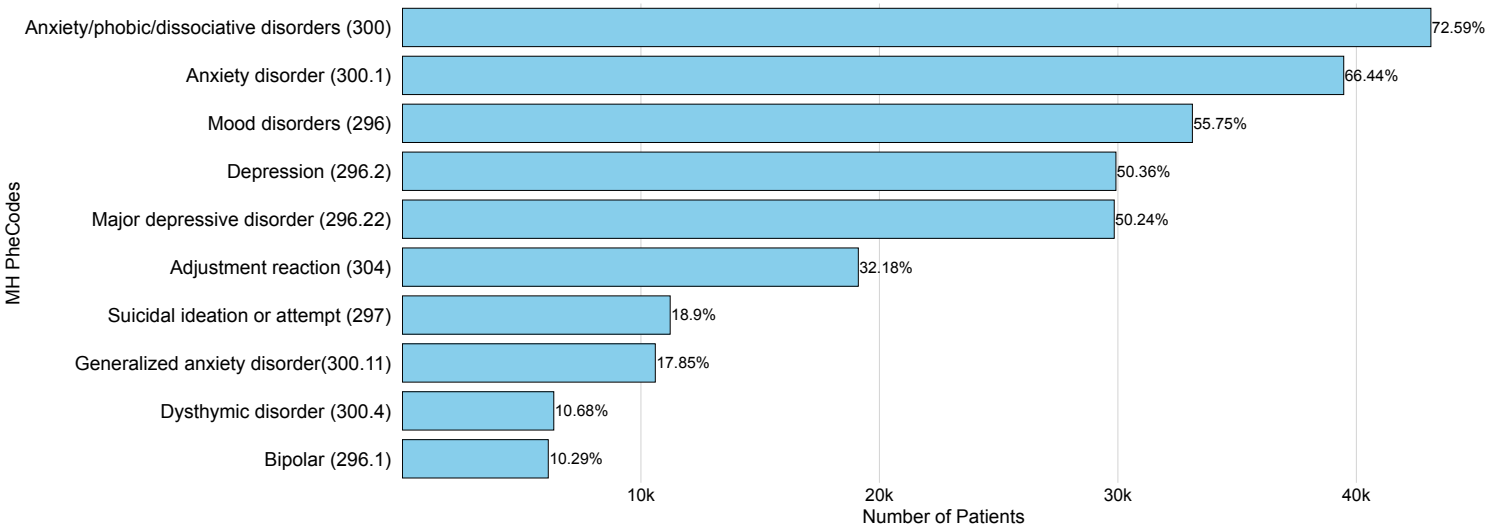


Figure 3.2: Prevalence of MH Conditions Among Patients

Figure 3.3 displays the distribution of the length of medical data available per patient in years. On average, patients have 15 years of medical data history available, corresponding to an average of 206 visits and 336 unique codes. In total, we identified 1869 different diagnosis codes, 1694 medication codes, 263 procedure codes, and 6214 lab codes. This high-dimensional dataset is a challenge for models to handle effectively.
In an additional experiment, we also consider the CUI codes extracted from the medical records. Due to the vast number of CUIs available, we selected only those related to SRB from ARCH [6] and ONCE [7], resulting in a set of 1272 CUIs for analysis.

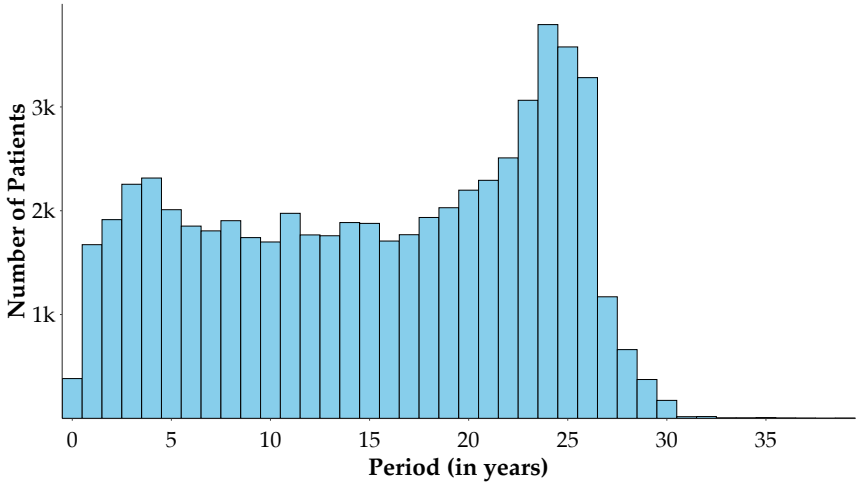


Figure 3.3: Distribution of Duration of Medical Data Available Per Patient in the Biobank dataset

RPDR Dataset

This dataset was used to pre-train the transformer models. Training the models on a large dataset before fine-tuning them on a smaller, task-specific dataset (Biobank), can enable the models to learn underlying patterns in EHR sequences, ultimately improving their ability to capture code interactions in the SRB prediction task.

The RPDR dataset contains data from 317,758 patients. Table 3.4 presents some statistics about this population. Similar to the Biobank dataset, the RPDR dataset mainly consist of females (65%) compared to males. Most patients are either married or single, and the majority are white. Regarding age, most patients are between 35-55 years old and the overall SRB prevalence in the dataset is 13%.

Figure 3.4 displays the distribution of the length of medical data history available per patient in years. On average, patients have 13 years of medical records, which represents an average of 106 visits and 194 unique codes per patient in our dataset. This is notably fewer than the codes found in the Biobank dataset. In total, we identified 1866 distinct diagnosis codes, 1706 medication codes, 243 procedure codes, and no laboratory codes in this dataset.

Setting	Number	Percentage (%)
Gender		
Female	207,791	65.4
Male	109,952	34.6
Race		
Asian	8,460	2.7
Black	18,455	5.8
White	252,607	79.5
Other	37,059	5.8
Unknown	1,177	0.4
Marital Status		
Widow	16,232	5.1
Married	123,130	38.7
Divorced	23,832	7.5
Single	139,215	43.8
Other	15,349	4.8
Age		
18-35	62,098	19.5
35-55	96,590	30.4
55-70	79,715	25.1
70+	79,355	25.0

Table 3.4: RPDR Population Statistics

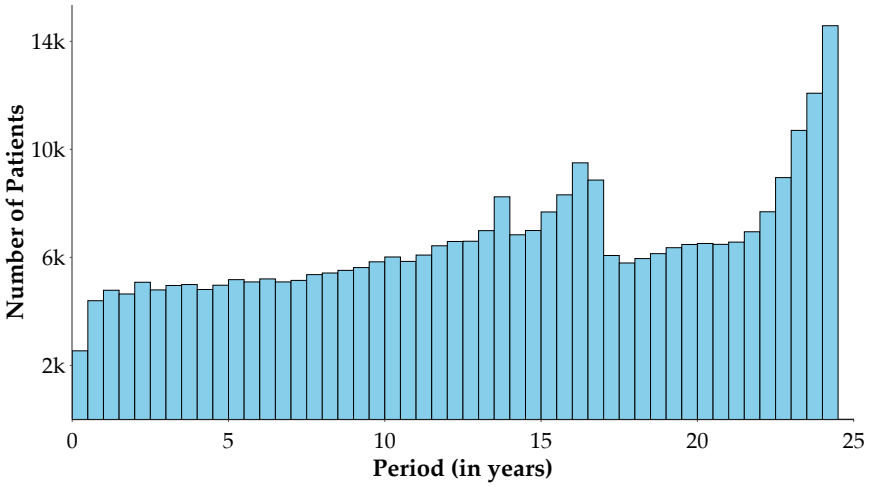


Figure 3.4: Distribution of Length of Medical Data Available Per Patient in the RPDR dataset

The discrepancies between the two datasets, particularly the absence of LOINC codes in the RPDR dataset, can be challenging during the pre-training of the model, depending on the chosen architecture. Indeed, this absence may restrict the model's ability to learn from laboratory results during pre-training. It is important to take this factor into account when selecting and designing the model architecture.

3.2 Landmark model - Regularized Cox Regression

3.2.1 Survival analysis

Survival analysis is a branch of statistics that focuses on analyzing time-to-event data. In medical research, it is often used to study the time to a particular event, such as death, the occurrence of a disease, or disease progression. In our case, the event of interest is the time to first SRB event since the landmark visit. Some key concepts in survival analysis include censoring, the survival function, and the hazard function.

Censoring

In survival analysis, censoring refers to the incomplete observation of survival times. Right censoring is the most common type of censoring, and occurs when a subject's survival time is only known to be greater than some value. This can happen when a subject is still alive at the end of the study, or when a subject is lost to follow-up during the study period.

For example, if we are considering a prediction window of two years, then all patients without SRB in the next two years after the landmark visit would be considered censored, even if they have an SRB events after two years.

Survival function

One of the key concepts in survival analysis is to estimate the probability of "surviving" longer than a certain amount of time. This is done by estimating the survival function, $S(t)$, which gives the probability of surviving beyond time t. The estimated probability $S(t)$ is a step function that changes value only at the time of each event.
The survival function is defined as: $S(t) = P(T > t)$ where T is the random variable representing the time to event.

Hazard function

The hazard function, $h(t)$, is another important concept in survival analysis. It represents the instantaneous risk of an event occurring at time t, given that the individual has survived up to time t. The hazard function is defined as:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = -\frac{d}{dt} \log(S(t))$$

Cox proportional hazards model

The Cox proportional hazards model is a semi-parametric model that models the relationship between the survival time and one or more predictor variables. It is built on the hazard function, and assumes that the hazard

function can be written as the product of a baseline hazard function, $h_0(t)$, and a function of the predictor variables, $\exp(\beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)$, where $X_1, X_2, \ldots, X_p$ are the predictor variables and $\beta_1, \beta_2, \ldots, \beta_p$ are the corresponding coefficients.
That is, the Cox Proportional Hazard Model fits the hazard rate

$$h(t|X_1, X_2, \ldots, X_p) = h_0(t) \exp(\beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p).$$

The "proportional hazards" part of the model name comes from the assumption that the hazard ratios between any two individuals are constant over time. The effect of a one-unit change in a predictor variable on the hazard is multiplicative and independent of time.

To estimate the coefficients $\exp(\beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)$, the Cox proportional hazards model maximises the partial likelihood given by:

$$L(\beta_1, \beta_2, \ldots, \beta_p) = \prod_{i=1, \delta_i=1}^{n} \left(\frac{\exp(\beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip})}{\sum_{j \in R(t_i)} \exp(\beta_1 X_{j1} + \beta_2 X_{j2} + \ldots + \beta_p X_{jp})} \right)$$

where n is the number of individuals in the study, X_{ij} is the value of the jth predictor variable for the ith individual, t_i is the observed survival time for the ith individual, δ_i is an indicator variable that is 1 if the ith individual experienced the event and 0 otherwise, and $R(t_i)$ is the set of individuals who are at risk prior to time t_i.
The reason why the partial likelihood is used instead of the full likelihood is that censored individuals contribute to the risk sets and thus influence the parameter estimates. The goal is not to predict censoring but to make assumptions about survival times and the effects of covariates, using all available data, whether it's fully observed or censored. A hazard ratio greater than 1 indicates an increased hazard (or shorter survival time), whereas a hazard ratio less than 1 indicates a decreased hazard (or longer survival time).

3.2.2 Model implementation

The landmark model is implemented using a random sampling approach to select one visit per patient to serve as the landmark visit. The date selected for the landmark visit must be neither the first month of the patient's collected data history nor the last month. Choosing the first month would result in an insufficient amount of observation data, while choosing the last month would make prediction and evaluation impossible. The following steps outline the implementation process:

Sample Landmark Visit: For each patient, we randomly select one visit from all their available visits, ensuring that there is at least one month of data available before and after the chosen landmark visit. Figure 3.5 illustrates that 90% of the patients have at least 6 months of medical history available before the landmark visit to train the model. 70% of the patients have at least 48 months of medical history, and 55% have more than 96 months.
After the landmark visit, 87% of the patients have at least 6 months of medical data, 78% have 12 months, and 70% have 18 months.

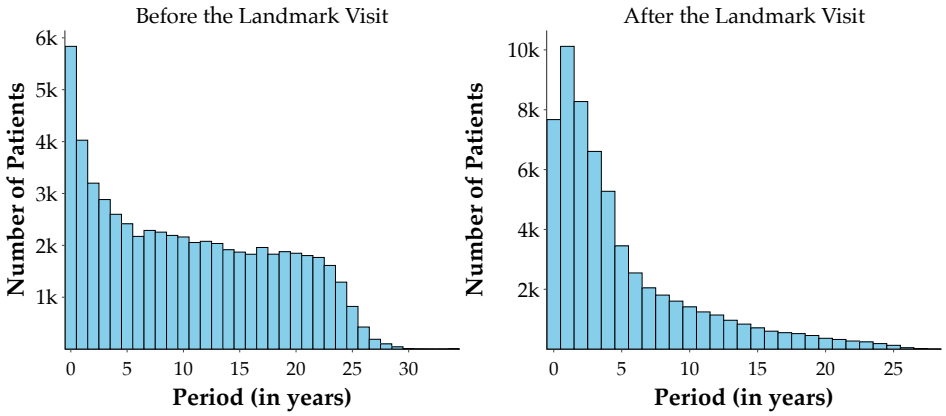


Figure 3.5: Distribution of Length of Medical Data Available Per Patient Relative to Landmark Visit

Compute Survival Outcome and status: We compute the time to the first SRB event following the landmark visit. If no SRB event occurs, the survival outcome is set to none. We also establish the patient's status: 0 if the patient is censored (i.e., no SRB event after the landmark visit), or 1 if the patient is potentially non-censored (i.e., at least one SRB event after the landmark visit). The final status and survival outcome will depend on the chosen prediction window. Table 3.5 provides information about the SRB prevalence for each prediction window. After the landmark visit, 8.2% of the patients experienced an SRB event. This proportion decreases to 2.3%, 3.1%, and 3.8% when considering prediction windows of 6, 12, and 18 months, respectively.

Prediction window	Prevalence
6 months	2.3%
12 months	3.1%
18 months	3.8%
∞	8.2%

Table 3.5: SRB Prevalence Across Multiple Prediction Windows

Build Landmark Dataset: For each look-back period, we create a corresponding landmark dataset. This dataset includes all data from the landmark visit, extending back to the specified look-back period (or all available data if the look-back period is shorter than the available data).
We then count the total number of codes per participant and remove rare codes that occur in less than 50 patients. We normalize each code count using the formula $log(phecodecount + 1) - log(T_{obs})$, where T_{obs} is the available look-back period and $phecodecount$ is the number of PheCodes for the given period.
Next, we calculate the survival outcome for censored patients as the duration in years between the last PheCode and the landmark visit. Additionally, we adjust the patient status if it was a non-censored patient (i.e., censor the patient if the event happens after the prediction window).
Lastly, we merge the dataset with the demographic dataset, which contains information such as the age at the landmark visit, gender, and race.

Survival Analysis: We conduct a survival analysis using a regularized Cox regression model that employs both L1 and L2 regularization.

The survival analysis is conducted across various look-back periods and evaluated on multiple prediction windows to gauge the model's performance under different scenarios.

3.2.3 Technical considerations

The model used for the analysis was a regularized Cox regression, leveraging the CoxnetSurvivalAnalysis from the python scikit-survival [39] library, which allows for the use of both L1 and L2 regularizations. The model was trained using a combination of Lasso and Ridge penalties, with a fixed numerical ratio of 9:1 and a minimum alpha value of 0.05.
To assess the performance of this model under different settings, multiple experiments were carried out. Cross-validation with K=5 folds was employed to evaluate the performance of the models.

3.2.4 Experiments

The purpose of re-implementing the landmark model was to assess its performance on our dataset and to establish a baseline for comparison with more sophisticated models. The landmark model [5] is a well-known model in the literature, but it only employs PheCodes. Consequently, we sought to explore the effect of incorporating additional types of codes, such as Medication, Procedures, and Lab codes, as well as CUI extracted from medical notes.

Adding such a large number of codes can significantly increase the number of features provided to the model. To address this challenge, we explored existing feature selection methods such as ARCH [6] and ONCE [7] to identify SRB-related features and therefore reduce the dimension of the feature space. We aimed to determine whether these methods could help us focus on the most relevant features for predicting SRBs and enhance the accuracy of our predictions.

Diagnosis code Only

We first re-implemented the setting used in the original Landmark model [5], using only the diagnosis code. As mentioned above, there are 1869 PheCodes in the Biobank dataset, of which only 35 (1.8%) are related to MH disorders (see table 3.1). However, these 35 MH codes account for 8% of the total count of PheCodes, indicating that they are relatively frequent compared to other codes.

Figure 3.6 shows the most common PheCodes for SRB patients, revealing both expected and unexpected findings. Well-documented risk factors for suicide, such as mood disorders, anxiety, alcohol-related disorders, and substance dependence, are consistent with commonly recognized conditions [16, 18, 19]. However, the inclusion of diagnoses such as hypertension and diabetes suggests a broader bio-psychosocial perspective, suggesting that chronic physical conditions may also significantly impact the mental health and risk profiles of individuals predisposed to SRB. This highlights the importance of a holistic approach to the assessment and management of at-risk patients, and it will be critical to analyze the contribution of these codes in the model.

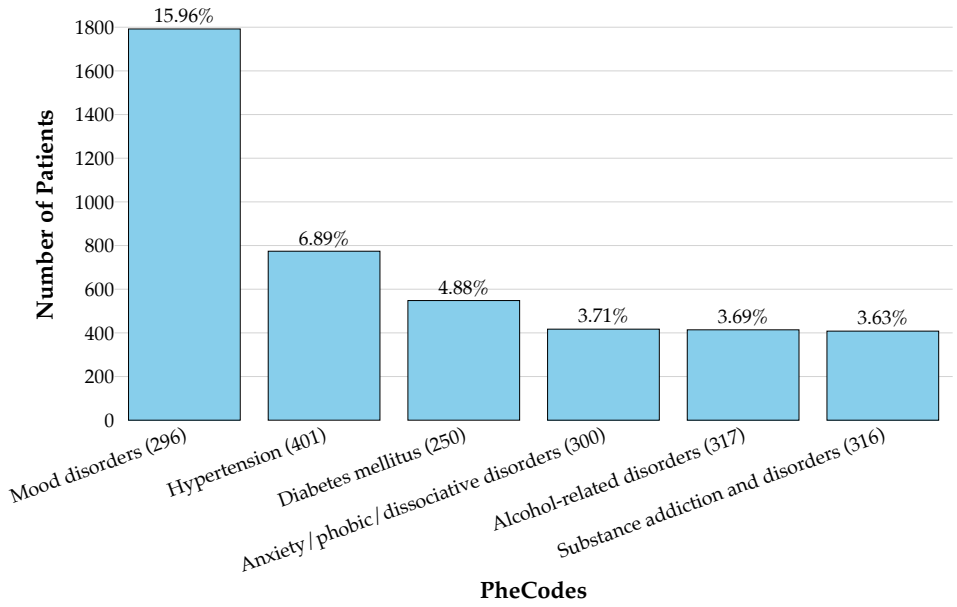


Figure 3.6: Most Frequent PheCodes for SRB patients

Combination of concept codes (Diagnosis, Procedure, Medication, Lab)

In addition to diagnosis codes, we included procedure, medication, and laboratory codes, resulting in a total of 10,040 codes. For each type of code, it is interesting to observe the most common codes for SRB patients and how they relate to the predicted outcome. These codes are shown in the table 3.6. For medication codes, we find mostly controlled substances such as oxycodone [40], clonazepam [41], or lorazepam [42], which are noteworthy because they treat conditions (such as severe pain, panic disorder, and seizure disorder) that are often associated with high levels of psychological distress. In addition, lorazepam has side effects such as paranoid or suicidal ideation. On the other hand, laboratory tests provide important biological insights that complement this picture. Tests for hematocrit [43], anion gap [44], and various plasma levels such as creatinine [45], carbon dioxide [46], and glucose [47] often reflect underlying physical conditions that can exacerbate mental health problems. For example, abnormalities in these tests can signal kidney dysfunction or metabolic imbalances, which have been linked to mood disorders and cognitive changes in the kidneys.

The aggregation of these diagnoses, medications, procedures, and laboratory markers can help identify patients who are at increased risk for suicide due to a combination of physical and mental health challenges. Thus, analyzing these codes together allows for a more holistic view of patient health, emphasizing the link between physical and mental well-being in suicide prediction and prevention.

Code	Description
Medication	
RxNorm 10582	Levothyroxine
RxNorm 6470	Lorazepam
RxNorm 7804	Oxycodone
RxNorm 5856	Insulin
RxNorm 2598	Clonazepam
Lab	
LOINC 4544-3	Hematocrit
LOINC 10466-1	Anion gap
LOINC 2160-0	Creatinine in plasma
LOINC 2028-9	Carbon dioxide in plasma
LOINC 2345-7	Glucose in plasma
Procedure	
CCS-PCS 218	Psychological and psychiatric evaluation and therapy
CCS-PCS 213	Respiratory intubation and mechanical ventilation

Table 3.6: Most Frequent Codes for SRB Patients

Combination of concept codes and CUI

When including CUIs, we initially considered expanding the SRB outcome to include not only PheCodes related to SRB (297, 297.1, 297.2), but also CUIs related to SRB, such as ”Suicidal” (C0438696) and ”Feeling suicidal” (C0424000). However, as shown in table 3.7, the prevalence of SRB CUIs was significantly higher compared to PheCodes due to the broader and more inclusive nature of CUIs. CUIs often encompass a wide range of terms and medical concepts, including various conditions related to suicidal behavior that may not be strictly classified under the PheCodes system. In addition, CUIs are extracted from medical records and may not directly reflect a patient's condition, but rather a discussion they had with their physician.

To minimize the risk of false positives (patients classified as SRB who are actually healthy), we decided to keep only the PheCodes related to SRB as an outcome.

Code	Description	Prevalence
PheCodes		
297	Suicidal ideation or attempt	18.8%
297.1	Suicidal ideation	7.2%
297.2	Suicidal or self-inflicted injury	1.8%
CUIs		
C0438696	Suicidal	31.3%
C0424000	Feeling suicidal	33.9%
C0038663	Suicide attempt	20.8%

Table 3.7: Prevalence of SRB Codes

Feature selection

Combining all concept codes resulted in a high-dimensional space of 10,040 codes. To focus on relevant codes related to SRB, we decided to use existing feature selection tools. We used the features selected from ARCH [6] and ONCE [7].
The ARCH knowledge graph uses a generative model to create embeddings from co-occurrence matrices that capture relationships between different medical concepts. Relevant codes are selected through a process that includes pre-screening by pointwise mutual information (PMI) testing and sparse embedding regression, which identifies significant relationships between medical concepts, thereby filtering out irrelevant ones.
On the other hand, ONCE generates a composite score for each feature based on relatedness to the target disease, frequency in EHRs, and weighted frequency in related articles, which helps to select the most relevant features for specific use cases.
These two methods allow the extraction of the most relevant features for specific conditions, such as suicide, and help reduce the total number of codes to 350.

Weight decay

To address the temporal dynamics of patient data, which is one of the critical limitations of the current model, which treats all historical events as equally relevant regardless of their temporal occurrence relative to a given landmark visit, we introduced weight decay. The intuition is that a medical code that occurs years ago is not expected to have the same impact on the model as one that occurs very close to the landmark visit.

To account for this, we employed a decay factor to modulate the impact of each code based on its timing. Specifically, the count for a given code c at visit i is calculated using the formula: $Count_{c^i} = d^{m_{c^i}}$, where d represents

the decay factor, set at 0.95, indicating the rate at which the importance of a code decreases with each month before the landmark visit. The variable m_{c^i} denotes the number of months between the code's recording and the landmark visit. Through this approach, codes recorded closer to the landmark visit are given more weight, reflecting their likely greater relevance to the patient's current health state, while those further in the past are progressively discounted.

3.3 Transformers

3.3.1 Technical description

The transformer architecture was introduced by Vaswani et al. [48]. It is a neural network architecture designed for sequence-to-sequence tasks, such as machine translation and text summarization. The transformer uses self-attention mechanisms to model the relationships between input and output sequences, without relying on recurrent neural networks (RNNs) or convolutional neural networks (CNNs).

Figure 3.7 displays the architecture of the transformer. It consists of an encoder and a decoder, each of which is composed of multiple identical layers. The encoder takes a sequence of input tokens and generates a continuous representation of the input sequence. The decoder takes the continuous representation of the input sequence and generates a sequence of output tokens.

Each layer in the encoder and decoder consists of two sub-layers: a multi-head self-attention mechanism and a position-wise feed-forward network. The multi-head self-attention mechanism allows the model to attend to different positions in the input sequence simultaneously, capturing long-range dependencies between tokens. The position-wise feed-forward network applies a non-linear transformation to each position in the input sequence independently.

The multi-head self-attention mechanism is the key innovation of the transformer architecture. It works by first projecting the input sequence into three different spaces: a query space, a key space, and a value space. Each position in the input sequence is represented by a query vector, a key vector, and a value vector. The attention weights are then computed by taking the dot product of the query vector and the key vector, and applying a softmax function to obtain a probability distribution over the input sequence. The output of the self-attention mechanism is then computed as a weighted sum of the value vectors, where the weights are given by the attention probabilities.

The transformer architecture also uses positional encoding to incorporate information about the position of each token in the input sequence. The positional encoding is added to the input token embeddings before they are fed into the encoder.

The output of the encoder is then passed through a feed-forward neural network and a softmax layer to generate a probability distribution over the output vocabulary. During training, the model is optimized to minimize the cross-entropy loss between the predicted probability distribution and the true output sequence.

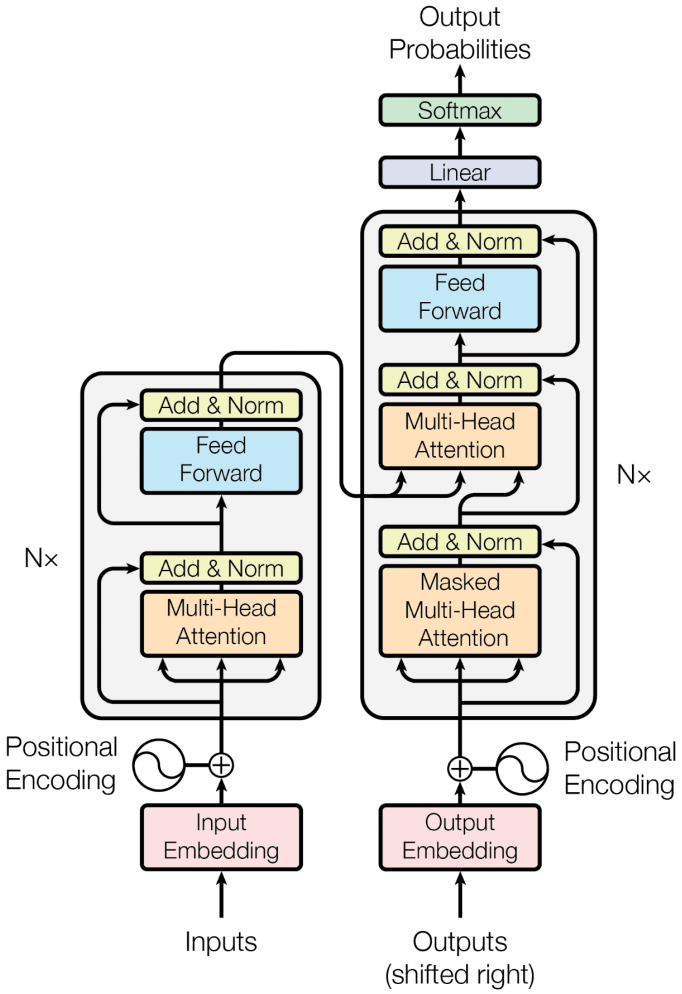


Figure 3.7: Transformer architecture

BERT

BERT [32] is a transformer-based model that learns contextualized representations of words in a sentence. BERT achieves this by using a bidirectional transformer encoder, which allows the model to capture information from both the left and right context of each word. The input to BERT is a sequence of tokens, which can be words, subwords, or characters. As seen in figure 3.8, each token is represented by a vector, which is the sum of three different embeddings: a token embedding, a segment embedding, and a position embedding. The token embedding represents the identity of the token, the segment embedding indicates whether the token belongs to the first or second sentence in a pair of sentences (for tasks that involve two sentences), and the position embedding encodes the position of the token

in the sequence. The embeddings are learned during pre-training and are fine-tuned for specific downstream tasks.

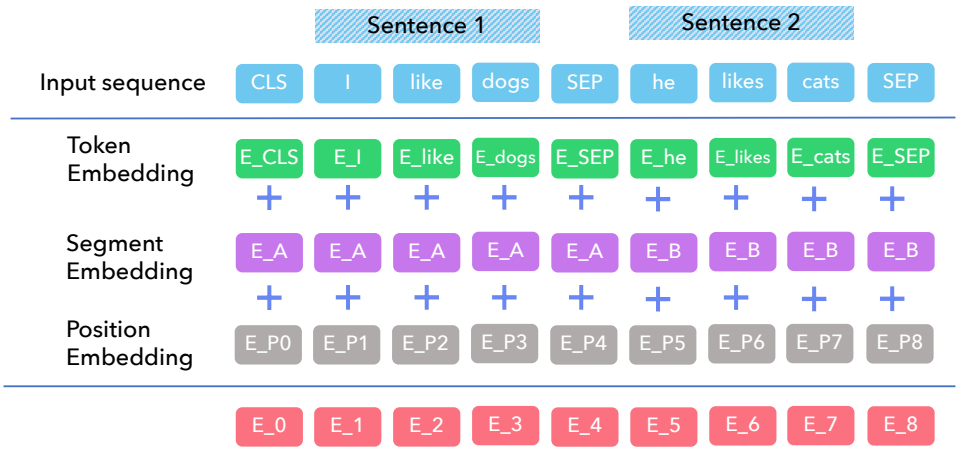


Figure 3.8: BERT Sequence Representation and Embedding. The subscripts show the original value for each embedding. The last row denotes the sum of the three embeddings as the output embedding.

The BERT architecture consists of multiple layers of transformer encoder blocks, described previously.
The key innovation of BERT is the use of a masked language modeling (MLM) objective during pre-training. In MLM, some of the input tokens are randomly replaced with a special [MASK] token, and the model is trained to predict the original tokens based on the surrounding context. This forces the model to learn rich, contextualized representations of words that can be used for a variety of downstream tasks.
In addition to MLM, BERT also uses a next sentence prediction (NSP) objective during pre-training. In NSP, the model is given two sentences and is trained to predict whether the second sentence follows the first sentence in the original text. This helps the model learn to capture relationships between sentences, which is important for tasks such as question answering and natural language inference.

During fine-tuning for specific downstream tasks, BERT uses a task-specific output layer on top of the transformer encoder. For example, for classification tasks, the output layer is a fully connected layer with a softmax activation function. The weights of the output layer are learned during fine-tuning, while the weights of the transformer encoder are fine-tuned from their pre-trained values.

In the context of EHR, transformers can be used to model patient sequences

of medical codes, such as diagnosis codes, medication codes, and procedure codes. By applying the self-attention mechanism to these sequences, transformers can capture complex relationships between the codes and predict clinical outcomes, such as the risk of readmission or the onset of a specific disease.

In the following sections, we will describe two transformer-based models for EHR analysis: BRLTM and Multiseq. Both models are based on the BERT architecture but have modifications to handle the specific characteristics of EHR data. We employ the same approach as the landmark model, using random sampling, look-back periods, and prediction windows. We perform two tasks: first, a classification task to predict whether the SRB event happens within the prediction window, and second, a survival analysis to predict risk scores and time-to-event. For the classification task, we use a binary cross-entropy loss function. To adapt these models for survival analysis, we slightly modify the loss function to a partial log-likelihood loss, which allows us to predict the time to the event of interest. By minimizing this loss function, the model learns to predict the time to event for each patient, taking into account censoring. This enables us to perform survival analysis on EHR data using transformer-based models.

3.3.2 BRLTM model

In this section, we describe the BRLTM architecture used in our first model. This model is based on the BEHRT [26] architecture and is presented in the paper Bidirectional Representation Learning from Transformers using Multimodal Electronic Health Record Data to Predict Depression [31]. We combined all types of codes, including diagnosis, medication, procedure, and lab codes, along with demographic information such as age, gender, race, and marital status.

Patient sequence representation

To prepare the data for BRLTM, we order the visits temporally. Two symbolic tokens are adopted: CLS denotes the starting point of the EHR, and SEP denotes the separation between two consecutive visits. The EHR sequence can be represented as: $V = [CLS, v_1, SEP, v_2, SEP, ..., SEP, v_{L_p}]$, where the visits $v_1, v_2, ..., v_{L_p}$, with L_p being the total number of visits for patient p, can be represented as:

$$v_i = [d_1^i, d_2^i, ..., d_{n_d}^i, m_1^i, ..., m_{n_m}^i, p_1^i, ..., p_{n_p}^i, l_1^i, ..., l_{n_l}^i]$$

Here, n_d is the number of diagnosis codes, n_m is the number of medication codes, n_p is the number of procedure codes, and n_l is the number of lab codes. It is possible to not have all types of codes within one visit, so some of these values may be zero. We maintain the order of the codes within a visit as follows: first diagnosis codes, followed by medication, procedure, and lab codes. The order of the codes within each type does not matter.

Data details

The vocabulary for this task combines all types of codes, resulting in a total of 3,815 codes for the RPDR dataset and 10,040 codes for the BioBank dataset. Using this sequence representation, we obtained relatively long sequences with an average length of 325 for the RPDR dataset and 1,830 for the BioBank dataset. This significant difference can be attributed to the fact that the RPDR dataset contains fewer codes (no lab codes) than the Biobank dataset, and the average number of codes per visit is significantly higher for the Biobank dataset. In addition, both datasets have an average of 13 to 15 years of medical history per patient and a health care utilization rate of 5 to 19 visits per year.

Embedding

In our BRLTM model, we treated each token embedding as a code embedding and extended the model's ability to aggregate demographics by adding

demographics embeddings such as age, gender, race, and marital status embeddings, as shown in Figure 3.9. There are four types of embeddings that are summed to generate the final output embeddings:

- Code embedding: It is composed of the four EHR data types mentioned in the patient sequence.
- Position and segment embedding: As in the original BERT model, position and segment embeddings indicate the position of one code in the full sequence, distinguishing codes in adjacent visits. However, as opposed to language where the order of a word in a sentence is highly important, in patient EHR sequences, the order of the code within a visit doesn't matter. Therefore, each code within the same visit has the same positional encoding. Annotating the position of each code/visit in the sequence enables the model to capture the positional interactions among EHR data. Segment embeddings are used to provide extra information to differentiate codes in adjacent visits by alternating between two trainable vectors, depicted as A and B in figure 3.9.
- Demographic embeddings: These are repeated in every position of the sequence. Combining code embeddings with the age embedding not only enables the model to use age information as a feature but also provides temporal information in the sequence.

Thanks to this representation and the BERT architecture, the model is able to aggregate multimodal EHR data into a single model and process them in a temporal manner, as well as investigating the inner association contingency between them in various visits.

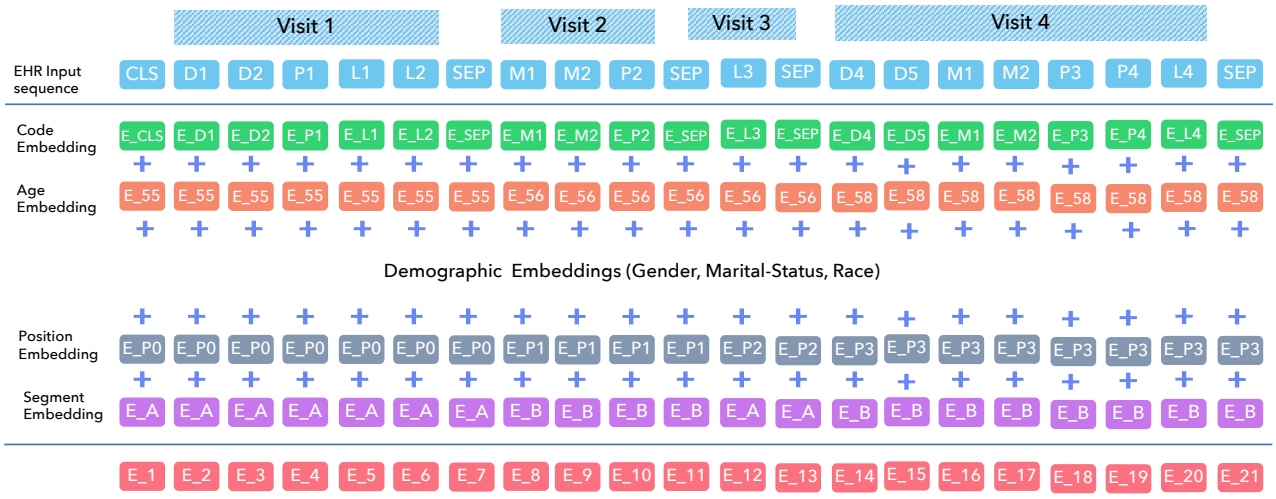


Figure 3.9: BRLTM Sequence Representation and Embedding. The subscripts show the original value for each embedding. D, M, P, and L denote diagnoses, medications, procedures, and labs, respectively. The last row denotes the sum of the seven embeddings as the output embedding.

3.3.3 Multiseq model

The BRLTM model has a limitation in that it cannot capture the irregularity between the visits, as it only uses age as a temporal feature. To address this, we incorporate temporal information in a similar way to CEHR [30] by inserting an artificial time token (ATT) between two neighboring visits based on their time interval. However, adding ATT tokens can result in longer sequences, which can be problematic. To mitigate this, we follow the approach of ExBERT [27] and group medical concepts into separate, learnable embeddings per concept type, rather than concatenating them into one long vector. This allows the model to learn which concepts to focus on and avoids exploding input lengths.

Patient sequence representation

The patient sequence is represented by four different sequences, one for each type of medical code (diagnosis, medication, procedure, lab). These sequences need to be aligned to be processed together by the model. For each visit, we count the number of codes of each type and take the maximum as an indicator of how many items per visit each sequence needs to have. If a visit has fewer codes of a particular type than the maximum, we pad the sequence to this maximum.

Let v_i denote the i-th visit, and let d^i, m^i, p^i, and l^i denote the sets of diagnosis, medication, procedure, and lab codes for that visit, respectively. We define the visit sequence v_i as follows: $v_i = [v_{d^i}^T, v_{m^i}^T, v_{p^i}^T, v_{l^i}^T]$, where each sequence is padded with a PAD token if necessary to ensure that all code sets have the same length.

Ex : Let's consider a visit i that has the following set of codes :
$d^i = \{d1, d2, d3, d4\}, m^i = \{m1, m2, m3, m4\}, p^i = \{p1, p2\}, l^i = \{\}$.

The maximum number of item in the visit i is given by :

$$n^i = max(len(d^i), len(m^i), len(p^i), len(l^i)) = len(d^i) = len(m^i) = 4$$

Therefore each concept code has a visit sequence:

$$v_{d^i} = [d1, d2, d3, d4], v_{m^i} = [m1, m2, m3, m4], v_{p^i} = [p1, p2, PAD, PAD], v_{l^i} = [PAD, PAD, PAD, PAD]$$

The final visit sequence will be noted:

$$v_i = [v_{d^i}, v_{m^i}, v_{p^i}, v_{l^i}]^T = \begin{bmatrix} d1 & d2 & d3 & d4 \\ m1 & m2 & m3 & m4 \\ p1 & p2 & \text{PAD} & \text{PAD} \\ \text{PAD} & \text{PAD} & \text{PAD} & \text{PAD} \end{bmatrix}$$

To incorporate temporal information, we insert an artificial time token (ATT) between two neighboring visits based on their time interval. The following logic is used for creating ATTs based on the following time intervals between visits:

- If less than 28 days, ATTs take on the form of W_n, where n represents the week number ranging from 0-3 (e.g., W_1).
- If between 28 days and 365 days, ATTs are in the form of M_n, where n represents the month number ranging from 1-11 (e.g., $M_1$1).
- Beyond 365 days, a long-term (LT) token is inserted.

In addition, we add two special tokens, V_S and V_E, to represent the start and end of a visit, respectively. A patient sequence (either diagnosis, medication, procedure, or lab sequence) can be represented as a list of visits, where each visit is represented as a concatenation of the four code sets and the temporal tokens:

$$V_p = [V_S, v_1, V_E, ATT_1, V_S, v_2, V_E, ATT_2, ..., V_S, v_{n_p}, V_E]$$

where n_p is the number of visits for the patient p.

Data details

In this model, we have a separate vocabulary for each concept code, as opposed to a unified vocabulary for all codes as in BRLTM. Thanks to this new representation, the sequence length can be shortened even with the introduction of Artificial Time Tokens (ATTs). Indeed, the average sequence length for the Biobank dataset is now 1,655 instead of 1,830. However, due to the composition of the RPDR dataset, the multiseq approach does not shorten the sequences, and the average sequence length is 370 instead of 325.

Embeddings

The Multiseq model follows the same embedding structure as the BRLTM model, with a few key differences. Instead of having one embedding for all codes together, there are now four different embeddings, one for each type of medical code. These embeddings are summed together with other demographic information, such as age, gender, race, and marital status. From a clinical perspective, it is also interesting to separate diagnoses, procedures, drugs, etc., as they have different clinical value for downstream applications. The embedding structure is shown in Figure 3.10.

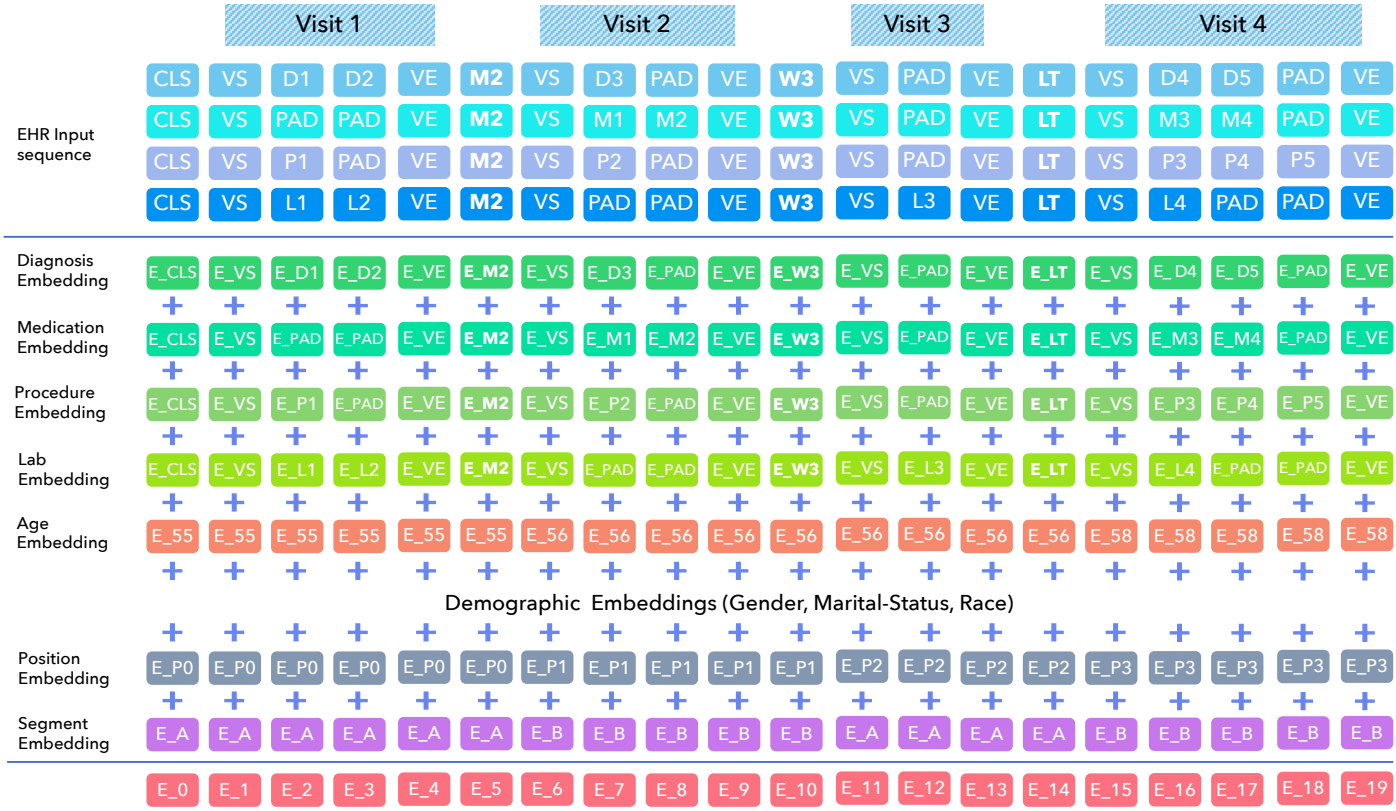


Figure 3.10: Multiseq Sequence Representation and Embedding. The subscripts show the original value for each embedding. D, M, P, and L denote diagnoses, medications, procedures, and labs, respectively. The last row denotes the sum of the ten embeddings as the output embedding.

3.3.4 Survival

In order to compare our models with the landmark model, which is a survival model, and to predict the time to event, we adapted our models for survival analysis. We followed the approach proposed in BERTSurv [36], where the Partial Log-Likelihood (PLL) loss function is modified to fit the neural network framework. The neural network is used to predict the hazard function, which represents the instantaneous risk of an event occurring at a given time. Instead of directly estimating the product of the regression coefficients and input features, $\beta^T X$, the network outputs $h_\theta(X)$, where θ represents the parameters (weights and biases) of the neural network, and X represents the input features, such as clinical codes embeddings, demographics, etc.

By minimizing the PLL loss, the model learns the parameters θ that best fit the survival data, enabling it to predict the time until the event of inter-

est (e.g., death) based on the input features. The PLL loss for a set of n individuals can be formulated as:

$$\text{PLLLoss} := -\frac{1}{n} \sum_{i:\delta_i=1} \left(h_\theta(X_i) - \log \left(\sum_{j \in R(T_i)} e^{h_\theta(X_j)} \right) \right)$$

where $\delta_i = 1$ if the event has been observed for individual i (uncensored), and $R(T_i) = \{j | T_j \geq T_i\}$ is the risk set at time T_i, containing all individuals who have not yet had an event or been censored before T_i.

This approach allows the model to leverage the power of transformers and deep learning to perform survival analysis, potentially improving the utilization of rich clinical data compared to traditional methods.

3.3.5 Technical considerations

Both models, BRLTM and Multiseq, were implemented in PyTorch and trained on a workstation equipped with a Tesla V100 GPU and 32 GB RAM. We followed the training scheduler with the Adam [49] optimizer used in the original BERT [32] model and set the warmup proportion and weight to 0.1 and 0.01, respectively. The Gaussian error linear unit (GELU) was used as the non-linear activation function in the hidden layers. Pretraining of MLM used the RPDR dataset with a minibatch of 32 patients for 100 epochs and was evaluated at every 20 iterations.

In all the experiments, cross-validation with K=5 folds was employed to evaluate the performance of the models for the finetuning of the prediction task. The models were trained with a minibatch of 32 patients for 20 epochs with early stopping. A dropout of 0.1 was set for both the hidden size and multi-head attention to address overfitting.

Table 3.8 provides some more detailed information about the models' implementation details.

	Pretraining	**Training**
Learning rate	1×10^{-4}	1×10^{-5}
Embedding Size	512	512
Attention layers	9	9
Attention heads	12	10
Intermediate layer	256	256

Table 3.8: Implementation details

3.3.6 Landmark Dataloader

We adapted the DataLoader to make it well-suited for the Landmark setting. When obtaining the EHR sequence, we randomly select one data point in the sequence that cannot be in the first 10% or last 10% of the data points. Additionally, at least 3 visits need to be available before the landmark visit. From there, we select the desired lookback period and compute the label and time to event according to the prediction window.

3.3.7 Experiments

Pretraining task - MLM

An EHR is composed of multimodal code sequences, analogous to the way language is composed of word sequences. Therefore, we hypothesized that the advantage of deep bidirectional sequential learning in language modeling over either a left-to-right model or the shallow concatenation of a left-to-right and a right-to-left model could be transferred to EHR modeling. Therefore, we adopted the same pre-training approach of MLM from the original BERT paper [32].

Specifically, we randomly selected 15% of the EHR codes and modified them using the following procedures: 80% of the time we replaced them with [MASK]; 10% of the time we replaced them with a random EHR code; and 10% of the time we left them unchanged. This structure in MLM forces the model to learn the distributional contextual representation between EHR codes, since the model does not know which codes were masked or which codes were replaced with a random code. Since only 1.5% of the codes are randomly replaced, the EHR modeling is not significantly affected. This random replacement introduces a small perturbation that distracts the model from learning the true contextual sequences of the EHR and forces it to identify the noise and continue learning the overall temporal progression.

We followed the precision score (true positives divided by predicted positives) at a threshold of 0.5 as the metric to evaluate the pre-training MLM task, which was the same approach as the BERHT model [26]. The average value is calculated over every masked code and over all patients.

For the Multiseq model, since we have one independent embedding for each concept code, we applied an MLM task separately to each embedding. The final MLM loss is therefore averaged over the MLM loss of each type of code. This approach allows us to leverage the detailed information contained in EHR data, enabling the model to learn nuanced representations for each code type, which can be crucial for tasks like suicide prediction requiring a deep understanding of medical records.

Table 3.9 presents the performance of the Masked Language Modeling (MLM) pretraining for the Multiseq and BRLTM models. It can be observed that the precision score increases gradually as the vocabulary size decreases by focusing on a unique data modality.

For the Multiseq model, we chose to combine the vocabulary from the RPDR and Biobank datasets, as each code type has its own embedding. This decision ensured that during fine-tuning, each embedding could learn from the pre-training and training datasets, especially in cases where no code was shared between the two datasets (e.g., lab codes). For the BRLTM, we used the RPDR dataset because all codes were combined. Thus, even if some lab codes are missing, the model is still able to learn from the other types of codes and their context.

	Vocabulary size	**Precision**
Multiseq		
Diagnosis	1895	0.76
Medication	1731	0.79
Lab	6236	0.97
Procedure	289	0.88
BRLTM		
All	3823	0.47

Table 3.9: Pretraining Performances

With and without pretraining

As previously discussed, pre-training, particularly with the MLM task, helps the model understand and predict the context of codes in EHR sequences. By randomly masking out words and attempting to predict them based on surrounding context, the model learns a deep, nuanced understanding of EHR sequence structure and code relationships. Since the model has learned some underlying patterns in the data, some transfer learning is possible, and models can be fine-tuned on a downstream task with a smaller dataset.

Pre-training can be advantageous in many tasks, but in cases where the vocabulary differs significantly between the pretraining and fine-tuning datasets, the benefits might be less pronounced. In our context, the RPDR and BioBank vocabularies are quite similar, except for the lab codes not present in the RPDR dataset. Consequently, we aimed to explore the impact of pretraining on each BRLTM and Multiseq model.

Feature selection

EHR sequences contain a rich number of information about a patient's health status. To improve model performance, we wanted to explore the impact of feature selection by removing irrelevant codes and focusing on informative codes related to suicide. As seen previously, some codes are more frequent in SRB patients, so identifying and prioritizing these codes can be beneficial. Since we are pretraining the models from scratch, selecting codes representative of the suicide task can improve the downstream SRB prediction task. This selective approach can also be applied to building the pretraining dataset, ensuring that the model learns from the most relevant contexts and interactions.

Moreover, the models have a maximum input length (512 tokens), and the EHR sequences are quite long, resulting in lost information. Feature selection can help optimize the sequence length, capturing more relevant information related to the SRB outcome. To achieve this, we decided to select features that appear to be "important" by the regularized Cox model presented earlier. This model has been used to predict the risk of SRB on the same dataset, and its regularization makes it a suitable first feature selector.

As shown in Table 3.10, this feature selection approach significantly impacted the vocabulary size and, consequently, the sequence length. The sequence length decreased from 1830 to 413 for the BRLTM Biobank representation and from 1655 to 508 for the Multiseq model. Interestingly, in this setting, the Multiseq sequences are longer than the BRLTM sequences.

In addition, the accuracy of the Multiseq pre-training improved significantly as the vocabulary was reduced. BRLTM accuracy also improved, but not as much as Multiseq. The model performs less well in predicting masked codes when all types of codes are aggregated than when only one type of code is focused.

	Vocabulary size	**RPDR sequence length**	**Biobank sequence length**	**Pretraining Precision**
Multiseq				
Diagnosis	35	157	508	0.93
Medication	34	-	-	0.94
Lab	53	-	-	0.97
Procedure	32	-	-	0.91
BRLTM				
All	70	152	413	0.52

Table 3.10: Feature Selection Sequences and Pre-training Performances

Chapter 4

Results

4.1 Metrics

The metrics used to evaluate the models are :

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**. The Receiver Operating Characteristic (ROC) curve is a graphical representation that summarizes the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for a binary classifier.
  In the context of suicide prediction, the ROC curve uses the predictive probability P_{SRB} as a risk score to distinguish between actual positive (SRB=1, indicating suicide risk) and negative (SRB=0, indicating no suicide risk) events. The predictions P indicate a positive result of the study if they are higher than a given threshold t.
  The ROC curve is then obtained by plotting, at different thresholds t, the TPR on the y-axis:

  $$y = P(P_{SRB} > t | SRB = 1),$$

  and the FPR on the x-axis:

  $$x = 1 - P(P_{SRB} < t | SRB = 0).$$

  The Area Under the ROC Curve (AUROC) is a performance metric used to evaluate the overall ability of the model to distinguish between the positive and negative classes. The AUROC takes values between 0 and 1, where a value of 1 indicates a perfect classifier. In the context of suicide prediction, a higher AUROC implies a better ability of the model to distinguish between patients with and without suicide risk.

- **Positive Predictive Value (PPV)**, also known as precision. It measures the proportion of true positive predictions (correctly identified suicidal

individuals) among all positive predictions (both correctly and incorrectly identified suicidal individuals). A high PPV means that when the model predicts a patient to be at risk of suicide, it is more likely to be correct. In the context of suicide prediction, a high PPV is crucial to minimize false alarms.

- **Sensitivity**, also known as recall or true positive rate. It measures the proportion of actual positive cases (suicidal individuals) that are correctly identified by the model. A high sensitivity means that the model is able to identify most of the patients who are actually at risk of suicide. In the context of suicide prediction, a high sensitivity is important to ensure that the model does not miss many at-risk individuals.

Threshold-dependent metrics, such as PPV and sensitivity, are reported at a 95% specificity threshold. Specificity is the proportion of actual negative cases (non-suicidal individuals) that are correctly identified by the model. A threshold at 95% specificity means that the model is set to prioritize the correct identification of non-suicidal individuals, allowing for a false positive rate of no more than 5%. In the context of suicide prediction, maintaining a high specificity is important to avoid unnecessary interventions for patients who are not at risk, while still identifying those who are truly at risk.

4.2 Landmark model

In this section, we present the results obtained from the landmark model under different configurations, as previously described. These include: using only diagnosis codes, using all concept codes (diagnosis, medication, procedure, and lab), integrating all concept codes with the CUIs, and using only codes selected by ONCE [7] or ARCH [6].

As shown in Figure 4.1, the model has good discrimination metrics (AUROC) across different prediction windows, with AUROC typically higher for shorter prediction windows. For example, when all concept codes are combined, the AUROC is above 0.855 for a 6-month prediction window, between 0.838 and 0.860 for a 1-year prediction window, and below 0.842 for an 18-month prediction window. This pattern is also observed for the sensitivity metric. In contrast, at 95% specificity, the highest PPV is observed using an 18-month prediction window.

In general, the performance of the model improves when using a longer look-back period, from 3 months to 48 months, and then remains relatively constant or slightly decreases when including more data history.

When comparing the different experiments, models that combine all concept

codes (diagnosis, medication, procedure, and lab) and those that combine all concept codes and CUIs significantly outperform models that use only PheCodes or a reduced number of codes in terms of AUROC. However, the results are more nuanced when other measures are considered. The model using CUIs does not necessarily outperform, and the model using selected features performs well.

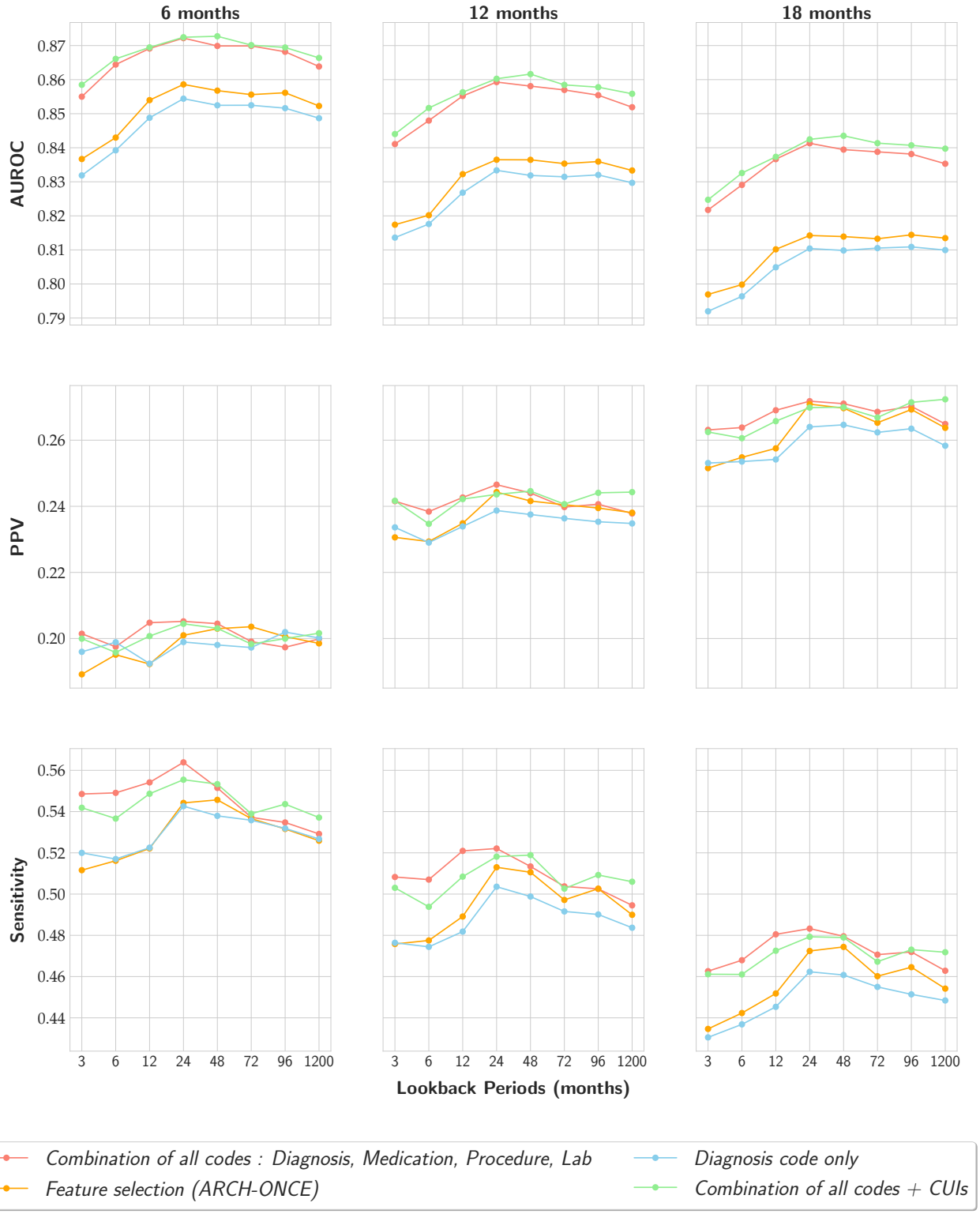


Figure 4.1: Cox Model Performance Over Each Look-Back Period and Prediction Window

Comparing the "classical" model with the model using weight decay in figure 4.2, we observe that for shorter prediction windows, the weights seem to help discriminate when the look-back period is long. However, for longer prediction windows, the discriminative performance of the model is lower when using this decay. When examining the other metrics, such as PPV and sensitivity, the application of weight decay appears to have a significant positive impact on performance as the look-back period increases.

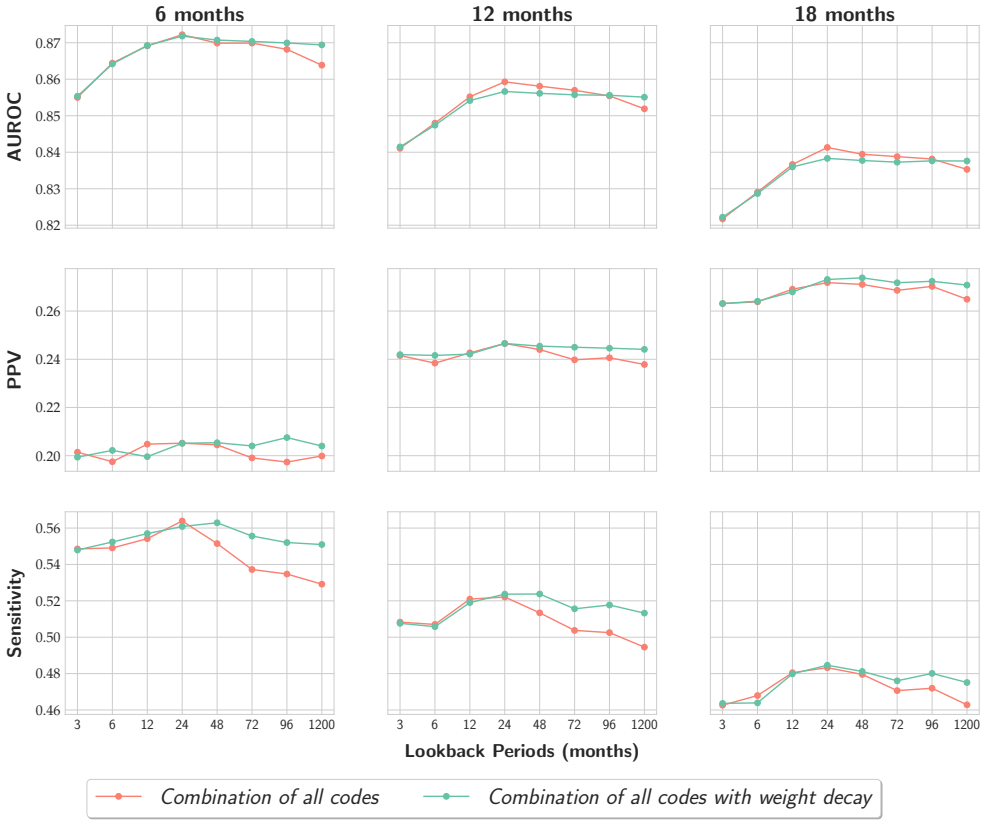


Figure 4.2: Cox model performances across each look-back period and prediction window

Figure 4.3 shows the top 15 features by absolute value of beta coefficients for the regularized Cox model in four different settings: using only diagnostic features, applying feature selection, combining all concept codes, and combining all concept codes with CUIs. The results shown are obtained with a 4-year look-back period (the top features were similar for other look-back periods). Since the features are standardized, the beta coefficients are directly comparable. We find codes such as PheCode 297 (suicide attempt or ideation), 296 (mood disorder) in all the different experiments. Demographic information such as marital status or gender is also shown for all

experiments.

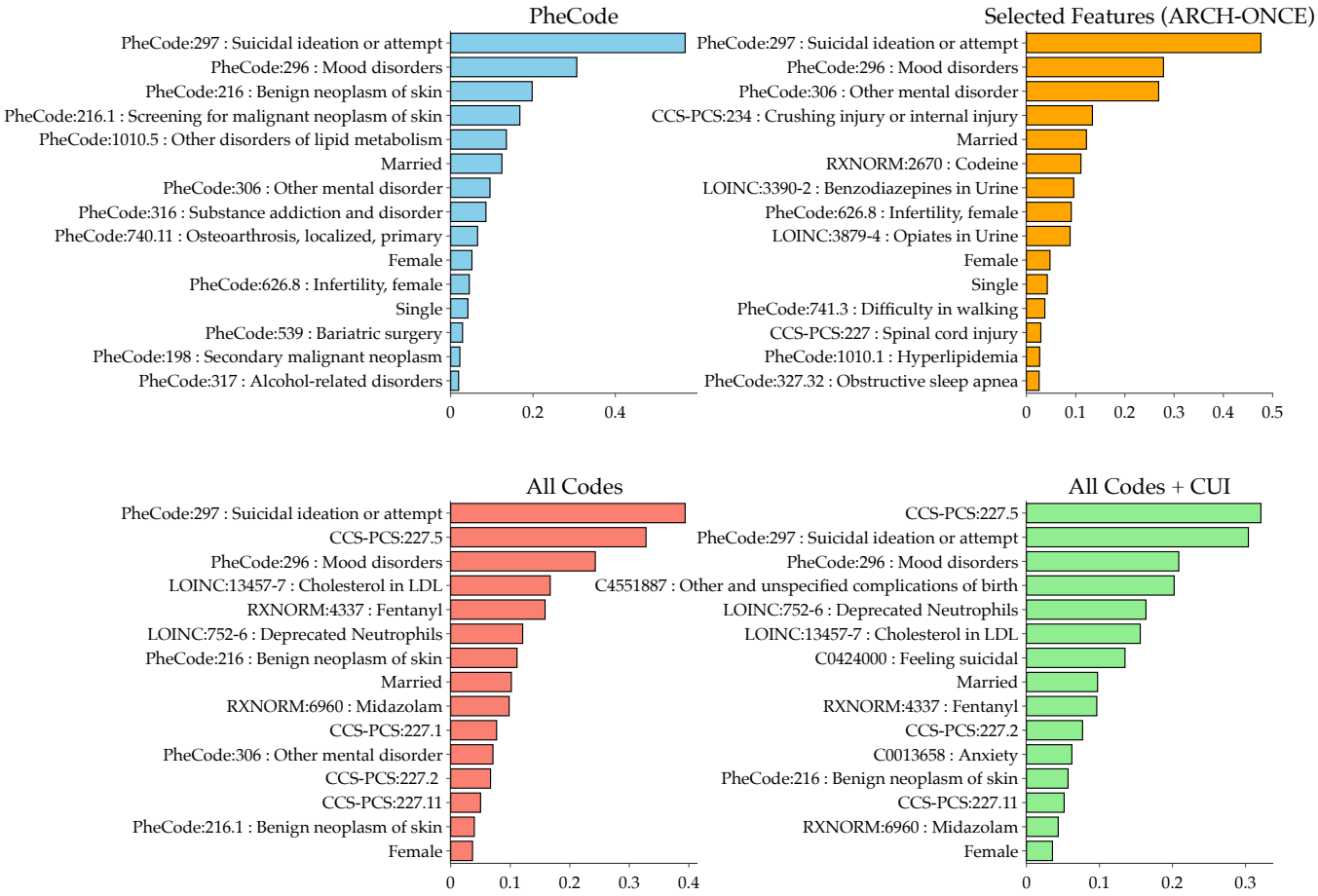


Figure 4.3: Top 15 Important Features for regularized Cox Model across different settings

By using regularization, the Cox model performs a form of feature selection by setting the coefficients of some codes to zero. The table 4.1 shows the number of ”important” features selected by the model across the different experiments. The number of features shown is a range, since more codes are present as the look-back period increases.

	Total number of features	**Important Features**
All concept codes	3228–5417	84–96
All concept codes + CUI	4192–6495	133–151
Only PheCodes	1409–1801	69–77
Feature selection (ARCH-ONCE)	276–329	63–71

Table 4.1: Number of Features by Experiments

To gain a deeper understanding of the feature selection results in ONCE [7] and ARCH [6], we compared the important features selected from the Cox model with those selected from ONCE and ARCH. All of the important features from the ARCH and ONCE were available in the Biobank dataset. However, 39 of the 71 important features from ARCH+ONCE model were not considered important by the classic model. Conversely, 57 of the 96 important features in the classic model were not included in ARCH or ONCE features. Table 4.2 lists the important features from the Cox model that were not included in ARCH and ONCE.

Code	Description	Code	Description
Diagnosis			
743	Osteoporosis, osteopenia and pathological fracture	216/216.1	Screening for malignant neoplasms of the skin
1010/1010.5	Other tests		
Medication			
10582	Levothyroxine	3264	Dexamethasone
1191	Aspirin	35629	Lactated Ringers
221147	Polyethylene Glycol	36387	Sennosides
2551	Ciprofloxacin	4337	Fentanyl
26225	Ondansetron	6387	Lidocaine
Lab			
10466-1	Anion gap	26474-7	Lymphocytes in blood
13457-7	Cholesterol	26478-8	Lymphocytes Leukocytes in Blood
1968-7	Bilirubin	26484-6	Monocytes in Blood
2028-9	Carbon dioxide in Serum or Plasma	26499-4	Neutrophils in Blood
2514-8	Ketones in Urine	26511-6	Neutrophils leukocytes in Blood
26449-9	Eosinophils in Blood	3719-2	Lithium in Serum or Plasma
26450-7	Eosinophils Leukocytes in Blood	4548-4	Hemoglobin A1c in Blood
Procedure			
130	Other diagnostic procedures; female organs	197	Other diagnostic ultrasound
231	Other therapeutic procedures	220	Ophthalmologic and otologic diagnosis and treatment
198	Magnetic resonance imaging	233	Laboratory - Chemistry and hematology
202	Electrocardiogram	235	Other laboratory
227.1/227.11	Unknown description	227.2/227.5	Unknown description

Table 4.2: Important Features from the Cox Model not Included in ARCH or ONCE

4.3 Transformer models

As mentioned previously, EHR data present many challenges, including complex relationships between past, present, and future codes, irregular intervals between visits, and concept codes of different sizes. In this section, we discuss how the implemented models handle these challenges and their overall performances.

Table 4.3 shows the performance of three different architectures across different experiments. Most models show reasonable stability across metrics with relatively low standard deviation values. Comparing the BRLTM and Multiseq models, we observe that Multiseq leads to better results when no feature selection is applied. Interestingly, Multiseq does not seem to benefit from pre-training compared to the BRLTM model.

Both models see an increase in performance when feature selection is ap-

Experiment	AUROC	PPV	Sensitivity
Regularized Cox Regression	0.853 ± 0.011	0.224 ± 0.009	0.551 ± 0.016
BRLTM			
Not pretrained	0.839 ± 0.016	0.219 ± 0.010	0.525 ± 0.030
Pretrained	0.848 ± 0.020	0.225 ± 0.006	0.551 ± 0.019
Pretrained + Cox feat select	0.869 ± 0.016	0.244 ± 0.016	0.588 ± 0.031
Pretrained (feat select) + Cox feat select	0.877 ± 0.011	0.244 ± 0.015	0.590 ± 0.030
BRLTM Surv			
Pretrained (feat select) + Cox feat select	0.880 ± 0.007	0.250 ± 0.015	0.587 ± 0.024
Multiseq			
Not pretrained	0.856 ± 0.006	0.225 ± 0.010	0.552 ± 0.033
Pretrained	0.843 ± 0.016	0.215 ± 0.012	0.530 ± 0.021
Not pretrained - no ATT	0.836 ± 0.014	0.187 ± 0.018	0.521 ± 0.023
Pretrained + Cox feat select	0.857 ± 0.009	0.222 ± 0.017	0.552 ± 0.033
Not pretrained + Cox feat select	0.864 ± 0.008	0.220 ± 0.016	0.544 ± 0.007

Table 4.3: Model Performance Comparison for Various Experiments with a 48-Month Look-Back Period and a 6-Month Prediction Window

plied. In this case, BRLTM outperforms Multiseq. It is also noteworthy that the model seems to benefit from ATT, as its removal decreases the performance of the model.

Finally, when the PLL loss is applied to convert the best performing model, BRLTM with feature selection, to a survival model, this results in slightly better performance.

Due to the computational resources required to run these models, we decided to select the best performing models, BRLTM and BRLTM Surv with feature selection, and run only these models over the full range of look-back periods and prediction windows.

As shown in Figure 4.4, the models show significantly better discrimination and sensitivity when using up to two years of data history, after which the performance remains relatively stable as more data history is added.

Again, we observe an opposite pattern for the PPV metric, which decreases from 3 months to 2 years of data history and then remains stable.

4.4 Comparison

In this section, we compare the two approaches discussed previously: the regularized Cox regression model and the BRLTM transformer.

Figure 4.4 illustrates the performance of the transformer and baseline models based on the three metrics we considered. The first plot, showing the AUROC, indicates that the transformer model significantly outperforms the baseline model across all three prediction windows. This performance difference is consistent across look-back periods, but is most pronounced when considering at least two years of medical data history. A similar pattern is observed for the sensitivity metric.

When examining the PPV, the transformer model also shows better performance. However, it is noteworthy that the PPV of the baseline model remains more consistent across look-back periods. In contrast, the performance of the transformer model decreases when adding up to one year of medical history and then stabilizes.

Looking at the ROC curves 4.5, the BRLTM curve remains above the Cox curve for all prediction windows. The shape of the curves for the 1-year and 1.5-year prediction windows are more similar than those for the 6-month prediction window. For a short prediction window, the BRLTM begins to exceed the Cox curve when the TPR is above 0.5, and then both curves converge as the FPR approaches 0.6. On the other hand, when a longer prediction window is used, both curves remain very close until a TPR of 0.7 is reached, and then again when an FPR of 0.85 is reached.

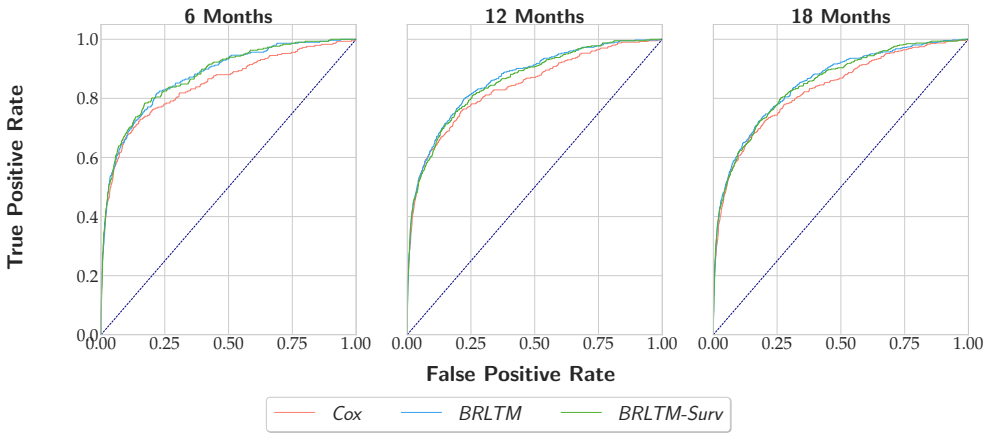


Figure 4.5: ROC Curves of BRLTM and Cox for a look back period of 48 months

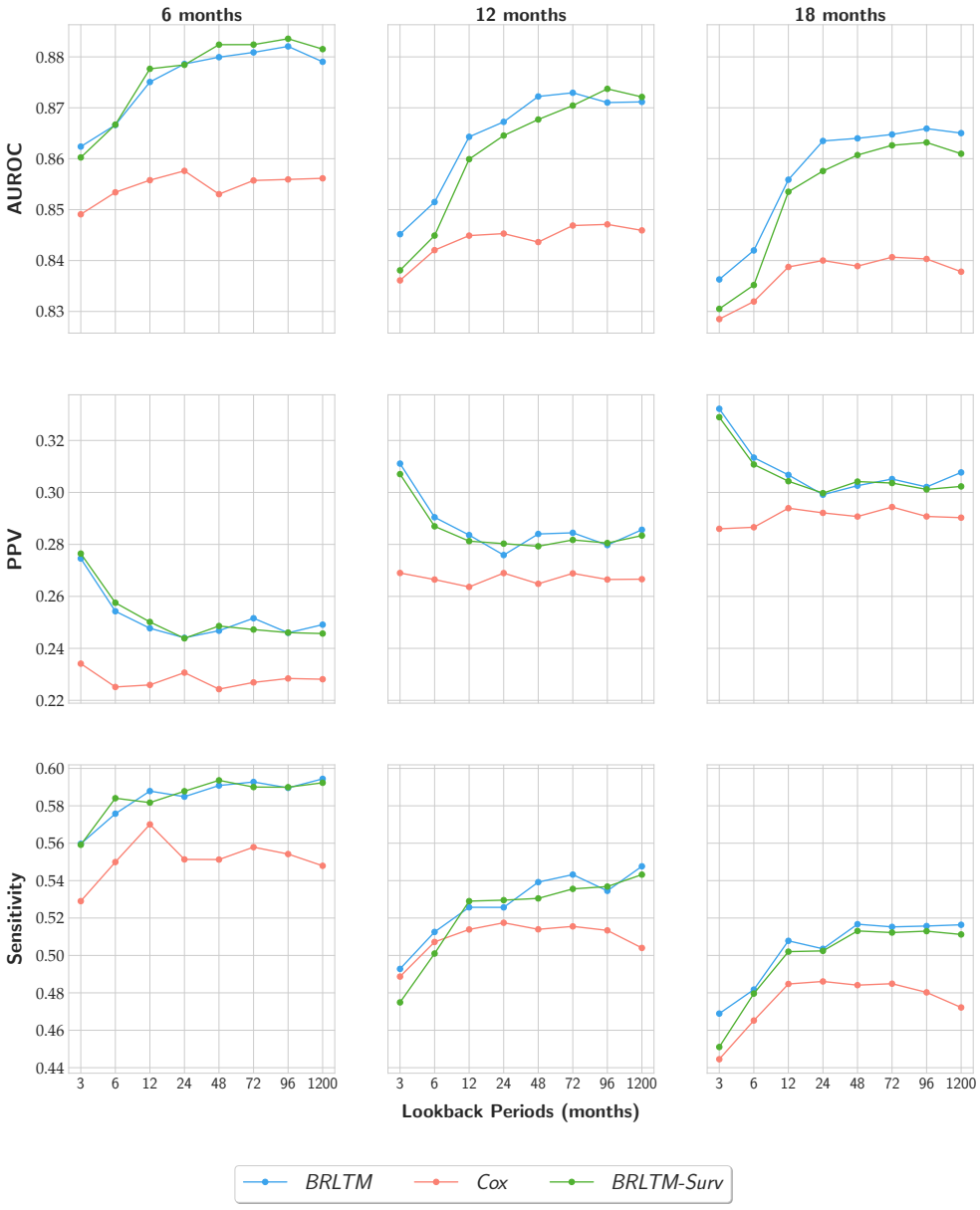


Figure 4.4: Cox and BRLTM comparison

Chapter 5

Discussion

We used large, structured EHR data to develop SRB prediction models using a landmark modeling approach that reflects real-world scenarios in which a physician would want to assess the risk of SRB. Both regularized Cox regression and Transformer models achieved high discriminative performance across all prediction window and look-back period combinations. This approach avoids the temporal bias in case-control sampling and provides an unbiased method for computing estimated risks across multiple clinical settings.

In this section, we discuss the results obtained for the different experiments and provide a comparison of the two models.

5.1 Regularized Cox regression

The landmark model offers valuable insights into the predictive capabilities of various combinations of clinical data across different prediction windows.

First, as shown in Figure 4.1, the predictive performance of the model appears to correlate positively with the length of the look-back period, suggesting that adding historical context to the model allows it to learn a more accurate representation of the patient. However, the slight decrease in model performance with an excessively long historical record raises the question of whether adding very old data adds more noise than relevant information to the model. From a clinical perspective, this suggests that SRB prediction relies primarily on recent data, but factors from many years earlier may also have a significant impact on the outcome.

When comparing different datasets, the combination of all concept codes significantly outperforms models based on PheCodes only in distinguishing cases from controls (AUROC). This demonstrates the value of combining different types of clinical data. In fact, the combination of All Codes and the

addition of CUIs tends to have the highest AUROC across most look-back periods and prediction windows, suggesting that using a multi-code type dataset maximizes the model's ability to discriminate between patients at risk and not at risk for SRB.

However, the combination of All Codes and adding CUIs does not necessarily translate to a significantly higher PPV compared to the other approaches. This suggests that the inclusion of irrelevant or less predictive features may be leading to false positives. The model using feature selection shows some results that are aligned with these approaches, indicating that more focused or streamlined data representations can enhance the model's precision in predicting true SRB cases.

Regarding the sensitivity metrics, the combination of all codes and also adding CUIs outperform other approaches for short look-back periods up to two years. However, when increasing the look-back period even more, the performance of these models drops and becomes more aligned with the feature selection and the model using only PheCodes. By adding old information in the model using a combination of codes, we are increasing the number of unique codes per patient and therefore highly increasing the dimension of the feature set. This can prevent the model from focusing on the relevant information for detecting SRB events.

The nuanced performance across PPV and sensitivity metrics, particularly when integrating CUIs, suggests that not all additional data types contribute equally to predictive performance and may introduce noise in some cases. However, due to the nature of the model, we are not able to capture the contribution of each code over time.

The regularization process in the Cox model, highlight the importance of certain clinical features over others. This suggests that the use of feature selection strategies has the potential to improve model performance. However, the performance of the model on the features selected by ARCH [6] and ONCE [7], which are knowledge graphs trained on a wide range of EHRs, shows that this is a delicate issue. A feature selection tailored to the specific task of SRB prediction is needed, and it may be beneficial to collaborate with clinicians on this task. As shown in table 4.2, many important features for SRB prediction are not included in the ARCH and ONCE features.

Finally, the weight decay model performs better when the forecast window is shorter, which intuitively makes sense since recent events are more likely to have an impact on an event that will occur in a few months. However, if we are interested in knowing whether someone will experience an SRB event in a year or more, then we may need to consider more historical data to explain it. As a general trend, we can see that when the look-back period is very long, at least 96 months, the weight decay improves the performance by helping to reduce the noise. This weight decay mechanism allows for

a more nuanced understanding of a patient's health evolution, providing a robust framework for modeling that captures the dynamic nature of medical histories. It also highlights the potential for exploring more complex models that could capture the evolution of code counts over a patient's lifetime.

When looking at the most important features in figure 4.3 across the different settings, we can see that some PheCodes such as Suicidal ideation or attempt (297) appear in each of them as the first or second most important factor, followed by mood disorders (300). We also find PheCodes related to alcohol disorders and substance addictions, which were some of the most frequent PheCodes in SRB patients (see figure 3.6). However, concerning the other types of codes, none of the most frequent medications, labs, or procedures in SRB patients appear to be in the top 15 most important features for the model.

5.2 Transformers models

As mentioned earlier, EHR data present many challenges, including complex relationships between past, present, and future codes, irregular intervals between visits, and concept codes of different sizes. In this section, we discuss how the implemented models address these challenges and their overall performance.

By comparing the Multiseq and BRLTM models, we notice a significant difference when using pretraining. Since the pretraining dataset lacks lab codes, the architecture used by Multiseq may not benefit from pretraining because an entire embedding is dedicated to lab codes and is significantly different during pretraining and fine-tuning. Consequently, it is difficult for Multiseq to learn from the pretraining dataset due to its dissimilarity to the Biobank dataset. On the other hand, the BRLTM architecture combines all codes into one embedding, allowing it to more effectively handle the lack of lab codes and learn the underlying patterns from the RPDR dataset.

In the Multiseq model, the significant increase in AUROC with the integration of artificial time tokens (ATT) into the EHR sequence suggests that this representation can capture the irregularity between visits and that the temporal aspects of medical data are important for improving prediction accuracy. In addition, Multiseq achieves better performance than BRLTM when all codes are used, which could be due to the fact that the sequences are very long and therefore there is more information loss in BRLTM compared to Multiseq. The combination of multiple embeddings for each type of code with ATT effectively addresses the challenges of handling very long sequences and tracking visit temporality, which are essential factors in suicide prediction.

However, with feature selection, the BRLTM architecture seems to be more appropriate since the sequences are shorter, and therefore the padding and the ATT might be too frequent compared to the actual codes.

The improvements observed with feature selection further demonstrate the value of refining input data to improve model results. The EHR vocabulary is very large, and by selecting only the codes that appear important according to the regularized Cox regression model, the transformer can focus on the interactions between these codes without being overwhelmed by noisy information. When feature selection is applied to both pre-training and fine-tuning datasets, performance is further enhanced, suggesting that a more targeted approach to feature selection throughout the training process can significantly benefit the model.

Applying survival loss to the pre-trained BRLTM model with feature selection improves the model's discriminative ability and PPV. It allows the model to discriminate between individuals based not only on the likelihood of an event, but also on when that event might occur. In contrast, cross-entropy loss treats predictions as binary outcomes (event or no event) without regard to timing. In addition, survival loss encourages the model to prioritize risk stratification over time, which is more consistent with real-world scenarios where understanding the order and timing of events is critical. This stratification is more nuanced than the simple classification provided by cross-entropy loss, enhancing the model's ability to correctly identify and prioritize higher-risk cases.

However, when looking at the analysis over different look-back periods and prediction windows (Figure 4.4), the survival model is not necessarily superior to the classifier. For shorter prediction windows (such as 6 months), the survival model can predict outcomes more accurately because the temporal dynamics and immediate risk factors have a more pronounced effect, and the time-dependent nature of the model takes advantage of this information. On the other hand, the binary cross-entropy model maintain or slightly improve its performance over longer time periods due to its simpler objective of classifying whether an event occurs within the prediction window, regardless of when it occurs. The longer the prediction window, the more variables and uncertainties can affect the outcome, attenuating the immediate temporal information on which the survival model relies. For the other metrics, the performance is very similar and do not allow for any conclusions.

The plots reveal critical insights into the temporal dynamics of medical data. The model significantly performing better when using significant data history up to two years underscores the relevance of the interaction between past, present, and future medical codes. However, the stable performance when adding more data history reflects the evolving nature of patients' health status and the diminishing relevance of older medical information.

5.3 Comparison

The results of comparing the Cox Landmark Model and the BRLTM Transformer in the context of SRB prediction provide several lessons about how effectively different prediction approaches manage history data.

First, the Transformer model's superior performance in AUROC across all prediction windows indicates its better ability to discriminate between patients who are likely and not likely to have an SRB event. This improvement is particularly evident when the model uses at least two years of medical history. This suggests that the Transformer model is better to exploit complex, longitudinal patterns in health care data. The higher the AUROC, the more effective the transformer is at correctly classifying patients according to their risk level.

The same pattern is observed for sensitivity, showing the ability of the transformers to identify true positives - that is, to correctly identify patients who will experience an SRB event. This is critical for SRB applications where identification of at-risk individuals can allow for timely intervention. The increased sensitivity with longer medical history suggests that the inclusion of more historical data allows the model to detect subtle trends that may indicate an increased risk of suicide.

However, performance on PPV presents a more complex picture. The Transformer model has an overall better PPV than the Cox model, indicating that it is better at predicting who will attempt suicide when they say they will. There is a significant decrease in PPV as data history increases from 3 to 24 months. This may indicate that even though the Transformer model becomes better at identifying true positives (see sensitivity), it also becomes more sensitive to false positives as the amount of historical data increases. This decrease may be due to overfitting or to the model becoming overly dependent on patterns that are not necessarily predictive of future suicide attempts. In contrast, the PPV of the Cox model remains stable across different lengths of medical history, suggesting that while it may not capture complex patterns as well as the transformer, it can still maintain a consistent level of accuracy in its predictions. This may be beneficial in settings where false positives are particularly costly.

Chapter 6

Conclusion

In this study, we have successfully reimplemented and extended a regularized Cox Model using a landmark setting, aligning it with clinical practice. We have enhanced the model by incorporating a wider range of EHR data types, such as diagnosis, medication information, procedure, laboratory, and clinical notes. This extension allowed for a more comprehensive understanding of patient condition and emphasized the relation between physical and mental well-being in suicide risk assessment.

In addition to the Cox model, our main contribution is the development of a transformer-based approach, BRLTM, built upon the BEHRT architecture, which uses the temporality of EHR data. To address the challenges posed by EHR data, such as complex relationships between past, present, and future codes, irregular intervals between visits, and concept codes of different sizes, we developed a novel model named Multiseq. This model combines different approaches to enhance prediction capabilities and overcome the cited challenges.

Our results demonstrated the benefits of using multiple types of codes in the Cox model and showcased the potential of transformer models to outperform the regularized Cox regression model in predicting suicide risk. The transformer models' ability to handle the temporal sequence of EHR data and capture complex interactions within patient records provided a more nuanced understanding of patient condition evolution and risk factors associated with suicide.

This study also highlighted the challenges related to high-dimensional EHR data and the significant improvement achieved by applying feature selection. The discussion emphasized the importance of integrating multiple EHR concept data, the significance of temporal dynamics in modeling patient data, and the potential of these models to provide clinicians with insights for early intervention strategies.

In conclusion, this work represents a substantial advancement in the use of EHRs for suicide risk prediction. By comparing traditional and advanced machine learning models, we have demonstrated the superior performance of transformer models in understanding the complex nature of EHR data. To the best of our knowledge, this work is the first to apply transformers to SRB prediction and shows significantly improved performance over traditional models for a variety of prediction windows. Therefore, this research opens a path forward for future research in predictive modeling for mental health and underscores the importance of continuous innovation in ML methodologies that can enhance clinical decision-making and preventive strategies in mental health care.

Limitations of this study include the lack of patient-reported data and the reliance on a single feature selection strategy. Future work should focus on validating these models on another dataset, improving interpretability through the implementation of attribution methods, exploring alternative feature selection strategies, and leveraging the knowledge of physicians to enhance the models' performance.

By addressing these limitations and pursuing the suggested future work, we can further refine and improve the predictive capabilities of our models, ultimately contributing to better mental health care and more effective suicide prevention strategies.

Bibliography

[1] World Health Organization. Preventing suicide: A global imperative. http://www.who.int/mental_health/suicide-prevention/world_report_2014/en/, 2014.

[2] Centers for Disease Control and Prevention. Suicide. https://www.cdc.gov/suicide/, 2022.

[3] R. Farmer, R. Mathur, K. Bhaskaran, SV. Eastwood, N. Chaturvedi, and L Smeeth. Promises and pitfalls of electronic health record analysis. *Diabetologia*, 2018.

[4] Office of the National Coordinator for Health Information Technology. National trends in hospital and physician adoption of electronic health records. https://www.healthit.gov/data/quickstats/national-trends-hospital-and-physician-adoption-electronic-health-records.

[5] YH. Sheu, J. Sun, and H. et al. Lee. An efficient landmark model for prediction of suicide attempts in multiple clinical settings. *Psychiatry Research*, 2023.

[6] Z. Gan, D. Zhou, and E. et al. Rush. Arch: Large-scale knowledge graph via aggregated narrative codified health records analysis. *medRxiv*, 2023.

[7] Online narrative and codified feature search engine (once). https://shiny.parse-health.org/ONCE/#.

[8] L Sher, M Oquendo, and JJ. Mann. Risk of suicide in mood disorders. *Clinical Neuroscience Research*, 2001.

[9] A. Naguy, H. Elbadry, and H. Salem. Suicide: A précis! *Journal of Family Medicine and Primary Care*, 2020.

[10] L. Sher and M. Oquendo. Suicide: An overview for clinicians. *Med Clin North Am*, 2023.

[11] JJ. Mann. A current perspective of suicide and attempted suicide. *Annals of Internal Medicine*, 2002.

[12] Phenome wide association studies (phewas). https://phewascatalog.org.

[13] Rxnorm. https://mor.nlm.nih.gov/RxNav/.

[14] Clinical classifications software (ccs). https://hcup-us.ahrq.gov/toolssoftware/ccs/ccsfactsheet.jsp.

[15] Logical observation identifiers names and codes (loinc). https://loinc.org.

[16] J. B., C. E. Martin, and J. L. Pearson. Contact with mental health and primary care providers before suicide: A review of the evidence. *American Journal of Psychiatry*, 2002.

[17] HH. Goldman. Review of general psychiatry. *Lange Medical Books*, 2000.

[18] L. Sher. Preventing suicide. *QJM : An International Journal of Medicine*, 2004.

[19] JJ. Mann and CA. Michel. Prevention of firearm suicide in the united states: What works and what is possible. *American Journal of Psychiatry*, 2016.

[20] Y. Barak-Corren, V.M. Castro, S. Javitt, A.G. Hoffnagle, Y. Dai, R.H. Perlis, M.K. Nock, J.W. Smoller, and B.Y. Reis. Predicting suicidal behavior from longitudinal electronic health records. *American Journal of Psychiatry*, 2020.

[21] Q. Chen, Y. Zhang-James, E.J. Barnett, P. Lichtenstein, J. Jokinen, B.M. D'Onofrio, S.V. Faraone, H. Larsson, and S. Fazel. Predicting suicide attempt or suicide death following a visit to psychiatric specialty care: a machine learning study using swedish national registry data. *Swedish National Registry Data*, 2020.

[22] S.E. Cho, Z.W. Geem, and K.S. Na. Prediction of suicide among 372,813 individuals under medical check-up. *Journal of Psychiatric Research*, 2020.

[23] A. Pigoni, G. Delvecchio, and N. et al. Turtulici. Machine learning and the prediction of suicide in psychiatric populations: a systematic review. *Translational Psychiatry*, 2024.

[24] SM. Mitchell, I. Cero, AK. Littlefield, and SL. Brown. Using categorical data analyses in suicide research: considering clinical utility and practicality. *Suicide and Life-Threatening Behavior*, 2021.

[25] S. Graham, C. Depp, EE. Lee, C. Nebeker, X. Tu, and H-C. et al. Kim. Artificial intelligence for mental health and mental illnesses: an overview. *Current Psychiatry Reports*, 2019.

[26] Y. Li, S. Rao, and J.R.A. et al. Solares. Behrt: Transformer for electronic health records. *Scientific Reports*, 2020.

[27] M. Rupp, O. Peter, and T. Pattipaka. Exbehrt: Extended transformer for electronic health records. *International Workshop on Trustworthy Machine Learning for Healthcare*, 2023.

[28] L. et al. Rasmy. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 2021.

[29] Z. et al. Yang. Transformehr: transformer-based encoder-decoder generative model to enhance prediction of disease outcomes using electronic health records. *Nature Communications*, 2023.

[30] C. et al. Pang. Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks. *Machine Learning for Health*, 2021.

[31] Y. et al. Meng. Bidirectional representation learning from transformers using multimodal electronic health record data to predict depression. *IEEE journal of biomedical and health informatics*, 2021.

[32] J. et al. Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018.

[33] M. Wu, X. Yang, and Y. et al. Liu. Development and validation of a deep learning model for predicting postoperative survival of patients with gastric cancer. *BMC Public Health*, 2024.

[34] S. et al. Wiegrebe. Deep learning for survival analysis: A review. *Artificial Intelligence Review*, 2024.

[35] H. Kvamme, Ø. Borgan, and I. Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of machine learning research*, 2019.

[36] Y. et al. Zhao. Bertsurv: Bert-based survival models for predicting outcomes of trauma patients. *arXiv*, 2021.

[37] Mass general brigham biobank. https://www.massgeneralbrigham.org/en/research-and-innovation/participate-in-research/biobank.

[38] R. et al. Nalichowski. Calculating the benefits of a research patient data repository. *American Medical Informatics Association*, 2006.

[39] Scikit-survival. https://scikit-survival.readthedocs.io/en/stable/.

[40] MedlinePlus. Oxycodone. https://medlineplus.gov/druginfo/meds/a682132.html.

[41] MedlinePlus. Clonazepam. https://medlineplus.gov/druginfo/meds/a682279.html.

[42] MedlinePlus. Lorazepam. https://medlineplus.gov/druginfo/meds/a682053.html.

[43] MedlinePlus. Hematocrit. https://medlineplus.gov/lab-tests/hematocrit-test/.

[44] MedlinePlus. Anion gap. https://medlineplus.gov/lab-tests/anion-gap-blood-test/#:~:text=The%20anion%20gap%20measures%20the,blood%20isn't%20acidic%20enough.

[45] MedlinePlus. Creatinine. https://medlineplus.gov/lab-tests/creatinine-test/.

[46] MedlinePlus. Carbon dioxide. https://medlineplus.gov/lab-tests/carbon-dioxide-co2-in-blood/.

[47] MedlinePlus. Glucose. https://medlineplus.gov/lab-tests/blood-glucose-test/.

[48] A. et al. Vaswani. Attention is all you need. *Advances in neural information processing systems*, 2017.

[49] DP. Kingma and JL. Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, 2015.