Research manuscript No.
(will be inserted by the editor)

Health Outcome Predictions with Data Transformation and Machine Learning

Linh Nguyen · Hegler Tissot

Abstract Machine learning's integration into healthcare has significantly advanced our ability to analyze and predict complex datasets. However, challenges remain in efficiently processing and predicting health outcomes from sparse and heterogeneous patient records. This study aims to address these challenges by evaluating three different machine learning methods: XGBoost with a tabular dataset, LSTM with a time series dataset, and a hybrid XGBoost then LSTM method. We utilize a synthetic COVID-19 patient dataset to assess the performance of these methods. Our results indicate that the XGBoost model achieves the highest F1 score (0.5199) with the shortest training time (151.42 seconds), demonstrating its efficiency and effectiveness. The LSTM model, despite capturing temporal dependencies, shows a significantly lower F1 score (0.2781) and the longest training time (approximately 8.66 days), highlighting its computational inefficiency and overfitting issues. The hybrid method improves over the standalone LSTM model but still falls short of XGBoost's performance in both accuracy and computational efficiency. These findings underscore the potential of using 2D models like XGBoost to effectively replace more complex 3D models, potentially leading to broader application and faster implementation in clinical settings, thereby enhancing the ability to provide timely and accurate patient care.

Keywords machine learning · healthcare · XGBoost · LSTM · hybrid models · temporal data · predictive modeling · patient records · COVID-19 · computational efficiency

1 Introduction

Machine learning's integration into healthcare has significantly advanced our ability to analyze and predict complex datasets [24]. Despite these advancements, challenges remain, particularly in efficiently processing and predicting health outcomes from sparse patient records [4]. Traditionally, healthcare applications have faced significant challenges in managing heterogeneous and temporally irregular longitudinal data, leading to substantial computational demands and the loss of critical temporal information [10]. Temporal information, which captures changes over time like the progression of a patient's symptoms or responses to treatment, is crucial for understanding disease trajectories and making informed medical decisions.

In the context of predicting health outcomes from patient records, a major challenge is efficiently utilizing longitudinal data formats, which incorporate time as a dimension [16]. These features can include a variety of patient information such as demographic details, medical history, and lab test results, providing a comprehensive snapshot of each patient's health profile. In contrast, data in healthcare typically involves time. For example, the progression of a patient's symptoms over time can provide crucial insights into the trajectory of an illness and help forecast potential complications or recovery timelines, which might include timestamped entries of symptom severity, medication doses, and physiological measurements like blood pressure or glucose levels [23]. In this research, three-dimensional data refers to the complex interplay of multiple patients, various timestamps, and diverse features, creating a multidimensional matrix where each element represents a unique profile that varies over time and across different patient profiles. This structure allows for the analysis of temporal dynamics critical for understanding health trajectories and predicting outcomes. This complexity not only increases the difficulty of analysis but also prolongs the time required for training predictive models.

The utilization of deep learning has made significant advancements in managing these complexities. For instance, (a) recurrent neural networks (RNNs) to improve the detection of early heart failure from electronic health records (EHRs), showing that RNNs can effectively harness the three-dimensional nature of EHR data to predict health changes more accurately [7]; (b) deep learning methods, utilizing the entire raw EHR including temporal data formatted with Fast Healthcare Interoperability Resources (FHIR), can accurately predict multiple medical events across different centers without site-specific data harmonization [19].

By processing the EHR data, the latter method outperforms traditional models in predicting in-hospital mortality, readmission rates, length of stay, and discharge diagnoses, validated by over 46 billion data points from 216,221 hospitalizations. However, these algorithms, which deal with complex three-dimensional data, are computationally expensive and require more efficient processing techniques to ensure timely diagnosis for a vast number of patients and to manage the influx of data effectively.

Decision tree methods, like gradient boosting and random forests, have gained attention for their rapid capabilities in capturing patterns and processing large datasets to predict health outcomes from EHRs, with significant advancements reported in recent research. These methods, such as random forests and gradient boosting, are favored for their ability to improve prediction stability and reduce the variance of the models, which is crucial in the medical field where prediction reliability can significantly impact patient outcomes. These methods are particularly adept at handling diverse and imbalanced datasets common in healthcare, providing robustness against overfitting which is often a challenge with deep learning models [18]. Moreover, random forests and gradient boosting demonstrated their effectiveness in interpreting complex interactions within EHR data, where permutation tests enhanced the clinical relevance of the predictions [5]. Additionally, they are adept at handling diverse and imbalanced datasets common in healthcare, providing robustness against overfitting — a frequent challenge with deep learning models. XGBoost effectively integrates different types of decision trees to harness their strengths, leading to improved performance on complex tasks such as disease prediction and patient management. For instance, XGBoost has been shown to outperform traditional models in cardiovascular risk prediction, achieving notable improvements in predictive accuracy and increasing the area under the curve (AUC) compared to previous models [20]. Notably, a study demonstrates that popular deep learning models for disease prediction are not meaningfully better than simpler, more interpretable classifiers such as XGBoost [8]. However, the decision tree methods (including XGBoost) normally ignored the temporal references in the dataset, which was expected since, unlike deep learning, decision tree methods cannot take in temporal references.

To address the challenge of incorporating temporal information into the XGBoost model while avoiding the high computational demands of deep learning, we propose a preprocessing strategy that transforms traditional three-dimensional data into a more manageable two-dimensional tabular format. This method allows for the inclusion of temporal references in XGBoost without the significant time and resource requirements typically associated with deep learning methods. The transformation is executed by segmenting the continuous timeline of data into fixed-duration windows and cumulating each window into a single record, thereby preserving essential temporal characteristics while enabling the data to be formatted into a tabular structure. This structured method retains key time-dependent features within each window, facilitating the application of 2D/tabular machine learning algorithms that are not inherently designed to process longitudinal data. This transformation allows the use of two-dimensional machine learning algorithms like XGBoost, which are designed for 2D datasets and offer a faster, less resource-intensive alternative to models requiring 3D data, such as neural networks or Long Short-Term Memory (LSTM). The predictive probabilities generated by XGBoost for each window are subsequently used as inputs for an LSTM model. This strategy enhances overall computational efficiency—both in terms of processing time and memory usage—more effectively than directly applying LSTM on raw data while still leveraging temporal information.

By comparing the performance of traditional LSTM models with an adapted 2D approach on a synthetic dataset of patient records, we can explore whether simplifying data representation from complex 3D formats into less complex 2D tabular formats enhances predictive capabilities. This adaptation involves using one-hot encoding to convert the original multidimensional data into a 2D format, facilitating the application of machine learning algorithms traditionally not suited for multi-dimensional data. Our study investigates if this transformation maintains, improves, or reduces the efficiency and accuracy of disease prediction, potentially offering a more streamlined method for handling extensive healthcare datasets.

Preliminary results indicate that the tabular dataset method using standalone XGBoost achieves a promising balance between accuracy (F1 score of 0.52) and computational efficiency. While the standalone LSTM model preserves the dataset's temporal dynamics, its slow processing rate poses significant limitations. Our hybrid XGBoost-LSTM method, which captures extended health information over staggered time windows, shows potential (F1 score of 0.49), though it requires further optimization to enhance predictive performance.

In our study, the application of a moving time window to analyze patient records, inspired by methodologies employed in intensive care unit (ICU) settings, demonstrated limitations when scaled to a larger and more varied dataset. The methodology presented by Tsiklidis [23] successfully applies time-series classification using machine learning to predict patient risks in ICU settings, utilizing both static and dynamic patient information and shows high predictive accuracy in a controlled environment with specific trauma patient types. However, when adapting a similar strategy of staggering records into several time windows to our more extensive dataset, which included various patient interactions beyond the ICU, the model's performance declined. This suggests a potential scalability and adaptability limitation of the time-window method when applied to datasets with greater complexity, variability, and sparsity. The F1 score of 0.49 for the patient window analysis, compared to 0.52 for the XGBoost model without temporal segmentation, indicates that the time-window method may be less effective in contexts outside the ICU where data uniformity and patient conditions are less controlled. This highlights the need for more robust machine learning frameworks that can handle larger, more diverse datasets without compromising the predictive accuracy critical for effective healthcare decision-making.

These findings underscore the feasibility of using 2D models to effectively replace more complex 3D models, potentially leading to broader application and faster implementation in clinical settings, thereby enhancing the ability to provide timely and accurate patient care. By simplifying the data structure, we can achieve comparable or even superior predictive accuracy with significantly reduced computational resources.

2 Related Work

The exploration of machine learning techniques in the management of longitudinal EHRs has become a focal point in medical informatics research. The complex, multidimensional nature of this data poses unique challenges, which various studies have addressed through innovative approaches.

Longitudinal or three-dimensional data in healthcare typically includes temporal dynamics essential for understanding patient trajectories and outcomes. Traditional methods often struggle with this complexity, but recent advances have shown significant promise. For instance, RNNs have been demonstrated to be effective in using longitudinal EHRs for early detection of heart failure, highlighting the importance of capturing temporal patterns in health data [7]. Similarly, comprehensive deep-learning approaches have been employed to process raw EHR sequences, including time-stamped entries, to predict various clinical events across multiple centers [19]. These studies underline the potential of deep learning techniques in managing and extracting value from complex longitudinal datasets, discussing the adaptability of LSTM networks to integrate and learn from diverse data types present in EHRs, and emphasizing their potential in predictive modeling and patient management.

A review of LSTM networks for predicting life expectancy using electronic medical records highlighted the strength of LSTM in handling variable-length and irregularly sampled data, common characteristics of EHRs [3]. Despite LSTMs' ability to model complex temporal relationships, challenges such as computational demand and handling of missing data remain significant hurdles. Further, a study proposed a model integrating an LSTM-based autoencoder with dense weighted small spheres and large margins (LSTMAE-DWSSLM) for classifying imbalanced time series data, demonstrating the effectiveness of LSTM autoencoders in learning temporally dependent feature representations from unlabeled data [13].

A survey evaluated different machine learning strategies, including decision trees, SVMs, and ensemble methods like XGBoost, for their efficacy in interpreting EHR data [12]. While not exclusively focusing on LSTM, this survey provided critical insights into the trade-offs between model complexity and interpretability, a crucial consideration when employing models like XGBoost in healthcare. In terms of integrating machine learning with EHR systems, another survey reviewed deep learning techniques for predictive modeling [21]. The findings highlighted the integration challenges and the potential of deep learning to transform predictive models in healthcare, also discussing the emergence of hybrid models that aim to combine the strengths of various machine learning techniques to improve performance and manageability. Another study utilized a targeted learning approach with daily EHR data, emphasizing the impact of data granularity on inference accuracy in healthcare predictions [22]. This study explored how different coarsening intervals could affect outcomes, offering a novel perspective on handling large-scale health data with machine learning tools like XGBoost.

In the realm of EHRs, the adaptation of machine learning models to predict health outcomes from tabular data has shown considerable promise. Specifically, proficient application of non-neural network machine-learning techniques has been illustrated to tackle the challenges posed by COVID-19 health outcome predictions using longitudinal EHR data [9]. Techniques such as feature vector representation and ANOVA for initial feature screening were utilized to achieve high predictive accuracy. This study employed models like the Gradient Boosting Machine, AdaBoost, Random Forest, and K-Nearest Neighbor to create an ensemble learning framework that robustly predicted various health outcomes. These models processed 2D tabular data derived from synthetic veteran EHRs, demonstrating their effectiveness in handling features encoded from a wide range of medical conditions and interventions.

An XGBoost-based model enhanced with novel time-dependent features was developed to predict sepsis in ICU patients [15]. This study leverages the strength of XGBoost, an ensemble learning method known for its high performance and efficiency in handling large datasets with complex feature interactions. The incorporation of time-dependent features into the XGBoost framework allows the model to dynamically adjust to changes in a patient's physiological state, providing a real-time risk assessment of sepsis onset. Moreover, a supervised deep learning model was introduced for clustering EHR data, focusing on identifying clinically meaningful phenotypes for both outcome prediction and patient trajectory analysis [1]. This study showcased the potential of advanced clustering techniques and feature-time attention mechanisms to enhance the interpretability and effectiveness of EHR predictions.

The potential of knowledge distillation with XGBoost for ICU mortality prediction was investigated, blending deep learning insights with XGBoost's efficiency to enhance both predictive power and explainability [14]. This work underscores the emerging trend of hybrid models that seek to leverage the strengths of both deep learning and traditional machine learning techniques. The research also proposed an XGBoost-based model enhanced with novel time-dependent features for dynamic prediction in ICU settings. Their approach effectively incorporates temporal dynamics into the model, significantly improving the prediction accuracy for sepsis.

While 3D models capture the full scope of data temporalities, they are computationally intensive. 2D models, although less demanding, often overlook the temporal aspect, which can be crucial for accurate predictions. The challenge lies in balancing computational efficiency with predictive accuracy. Deep ensemble learning approaches in healthcare suggest that combining multiple models might offer a solution to leverage strengths and mitigate weaknesses of both data types [18]. Additionally, ensemble deep learning for biomedical time series classification provides insights into how ensemble methods can be tailored for time-sensitive data, potentially offering a pathway to integrate 2D efficiency with 3D data richness [11].

A recent comparative study on machine learning and deep learning approaches for EHR data extensively covered the performance of LSTM against other models like GRUs and XGBoost [2]. Their findings provide a useful benchmark for understanding the conditions under which each model excels and offer guidance for researchers choosing between these models based on the specific needs of their datasets. While they provided insights into the comparative performance of LSTMs, GRUs, and XGBoost, the literature still lacks comprehensive surveys that juxtapose these models in hybrid settings specifically tailored for longitudinal EHR management. This gap underscores the need for more detailed comparative studies and evaluations of hybrid models.

This review of related work indicates robust interest and ongoing research into the use of LSTM, XGBoost, and their integration into healthcare data. While studies demonstrate LSTM's effectiveness in capturing temporal dependencies and XGBoost's performance with structured data, a gap remains in comprehensive comparative studies and hybrid model evaluations. Existing research often focuses on short-term, dense datasets like those from ICU settings, overlooking more extensive, sparse, and lifelong patient records. These datasets, while valuable, do not encompass the broader and sparser longitudinal records found in lifelong healthcare data. Our research aims to fill this gap by comparing LSTM, XGBoost, and hybrid models in handling longitudinal EHR data. By examining these models in real-world, long-term datasets, we aim to provide insights into their scalability, efficiency, and predictive accuracy, offering a balanced solution that leverages the strengths of both approaches for more effective and timely healthcare interventions.

3 Materials & Methods

3.1 Dataset Description

The study utilizes the Synthea™ Novel coronavirus (C-OVID-19) synthetic dataset. This dataset comprises longitudinal electronic health records (EHRs) of 124,150 synthetic patients, simulating the progression and treatment of COVID-19 from March through May 2020. In this study, we focus on a subset labeled `Injury of heart` involving 18,177 hospitalized patients. The dataset features detailed event-based patient records, allowing for the investigation of disease progression over time [25].

Each patient record contains multiple data points over time, categorized into various health-related events such as medical history, demographic information, and clinical events. For instance, a single patient's record contains demographic details such as birth year, age, gender, race, and ethnicity. It also includes clinical events like medication administration, procedures, care plans, and vital signs and lab results observations. Each event is precisely timestamped and coded, providing a detailed chronology of the patient's health history. The data is organized into several key components:

1. Event Timestamps: Each event is timestamped, indicating the exact date and time when the event occurred. This allows for precise tracking of the sequence and timing of events throughout the patient's medical history.
2. Temporal Information: The dataset includes fields indicating the number of days since the patient's first recorded event (`days_fore`) and the number of days before the current date (`days_back`). This helps in understanding the temporal context of each event relative to the patient's entire medical history.
3. Demographic Information: The dataset captures essential demographic attributes such as birth year, age, gender, race, ethnicity, and county. These attributes provide context about the patient's background and are crucial for analyzing health disparities and outcomes.
4. Clinical Events: A wide range of clinical events are recorded, including:
   – Medications: Information on medication administrations, including the specific drug codes and names.
   – Allergies: Records of any allergies the patient has, along with the corresponding codes.
   – Procedures: Details of medical procedures undergone by the patient, identified by procedure codes.
   – Care Plans: Records of care plans established for the patient's treatment.
   – Immunizations: Data on immunizations received, including the types and dates.
   – Conditions: Diagnosed medical conditions, identified by condition codes.
5. Observations: The dataset includes both categorical and numerical observations related to the patient's health. Categorical observations might consist of information such as smoking status, while numerical observations capture vital signs and lab results, such as blood pressure, heart rate, and various lab test values (Table 1).

3.2 Predicting Heart Injury

Our experimental setup consists of three distinct methods to predict health outcomes from patient records using machine-learning techniques that leverage both tabular and longitudinal data formats. We aim to evaluate the effectiveness and efficiency of each method in handling longitudinal health data for predicting a specific health condition, namely `Injury of heart`.

3.2.1 Tabular Dataset with XGBoost

This method utilized a 2D tabular format for the dataset, focusing on demographic information and binary or continuous health event indicators. The data were prepared using one-hot encoding, and the XGBoost model was used for classification.

Methodology for Data Preparation

We apply one-hot encoding to convert the original multidimensional data into a 2D format. The dataset is pre-processed to create a structured tabular format. Each health-related event within the patient's history is converted into a distinct feature. Binary indicators are assigned to these features: "1" if the event occurred at least once within the observation window for a patient, and "0" otherwise. This binary transformation allows for direct utilization of categorical data within the XGBoost framework.

For non-categorical features, such as physiological measurements or lab results that are captured multiple times, we aggregate the data to include the average, minimum, and maximum values recorded during the observation period. These statistical summaries provide a comprehensive representation of the patient's condition over time and are included in the dataset as separate features (Figure 1).

The input data to the model is a two-dimensional tabular dataset. Each row represents a single patient,

Table 1: Example of One Patient Records

Event When `event_when`	Days Fore `days_fore`	Days Back `days_back`	Head Name `head_name`	Rela Name `rela_name`	Tail Name `tail_name`
3/13/69 0:00	0	18624	0abbf153-7241	rBirthYear	1969
3/13/69 0:00	0	18624	0abbf153-7241	vAge	53.8
3/13/69 0:00	0	18624	0abbf153-7241	rGender	female
3/13/69 0:00	0	18624	0abbf153-7241	rRace	white
3/13/69 0:00	0	18624	0abbf153-7241	rEthnicity	nonhispanic
3/13/69 0:00	0	18624	0abbf153-7241	rMaritalStatus	married
3/13/69 0:00	0	18624	0abbf153-7241	rCounty	Middlesex County
5/25/00 0:00	11396	7228	0abbf153-7241	rCondition	162864005
12/19/13 0:00	16352	2272	0abbf153-7241	rCondition	230690007
11/25/18 0:00	18154	470	0abbf153-7241	rMedication	1000126
3/13/19 0:00	18262	362	0abbf153-7241	vObservation_QOLS	1
3/13/19 0:00	18262	362	0abbf153-7241	vObservation_QALY	48.9
3/13/19 0:00	18262	362	0abbf153-7241	vObservation_DALY	0.1
3/14/19 0:00	18263	361	0abbf153-7241	rCondition	68496003
6/6/19 0:00	18347	277	0abbf153-7241	rProcedure	76601001
9/5/19 0:00	18438	186	0abbf153-7241	rProcedure	76601001
11/20/19 0:00	18514	110	0abbf153-7241	rMedication	1000126
3/9/20 0:00	18624	0	0abbf153-7241	rCareplanReason	840544004
3/9/20 0:00	18624	0	0abbf153-7241	rCareplanReason	840539006
3/9/20 0:00	18624	0	0abbf153-7241	rCareplan	736376001
3/9/20 0:00	18624	0	0abbf153-7241	rCareplan	736376001
3/9/20 0:00	18624	0	0abbf153-7241	rCondition	36955009
3/9/20 0:00	18624	0	0abbf153-7241	rCondition	840544004
3/9/20 0:00	18624	0	0abbf153-7241	rCondition	386661006
3/9/20 0:00	18624	0	0abbf153-7241	rCondition	84229001
3/9/20 0:00	18624	0	0abbf153-7241	rCondition	840544004
3/9/20 0:00	18624	0	0abbf153-7241	rProcedureReason	840544004
3/9/20 0:00	18624	0	0abbf153-7241	rProcedure	261352009
3/9/20 0:00	18624	0	0abbf153-7241	rObservation_80383-3	Not_detected
3/9/20 0:00	18624	0	0abbf153-7241	rObservation_80382-5	Not_detected
3/9/20 0:00	18624	0	0abbf153-7241	rObservation_94531-1	Detected
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_8462-4	86
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_8480-6	107
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_8867-4	129.9
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_9279-1	29.9
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_2708-6	78.8
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_29463-7	84.6
3/9/20 0:00	18624	0	0abbf153-7241	vObservation_8310-5	38.5

and each column represents a specific feature derived from the patient's records, such as demographic information and health events, converted into binary indicators or aggregated statistics (Table 2).

Predictive Modeling Techniques

The predictive model is constructed using XGBoost, a decision-tree-based ensemble machine-learning algorithm that uses a gradient-boosting framework. The model is specifically tuned for the binary classification task of predicting the occurrence of `Injury of Heart`, identified by the label `label_card_86175003_ih`.

Model Training and Evaluation

1. *Model Initialization and Training:* We initialize the XGBoost classifier with varying tree depths, exploring depths from 5 to 30 to determine the optimal complexity of the model. Each model configuration is trained using the training subset of the data.
2. *Threshold Optimization:* For each model configuration, the predicted probabilities for the tuning set are used to determine the optimal threshold for converting probabilities to binary predictions. This threshold is selected to maximize the F1 score, balancing the trade-off between precision and recall.
3. *Model Selection:* The model yielding the highest F1 score on the tuning set is selected as the best model. This process involves comparing the F1 scores across different tree depths and selecting the depth and corresponding threshold that results in the highest F1 score.
4. *Final Model Performance:* The final selected model is evaluated on the test dataset to assess its performance. This evaluation includes calculating the F1 score and generating a confusion matrix.

Table 2: Example of All Patient Records in One Table after One-Hot Encoding (One Patient Each Row)

patient	birth_yr	gender_male	county_norfolk	...	condition_230690007	medication_1000126	procedure_766601001	observation_qols_min	observation_qaly_min
0abbf153-7241	1969	0.0	1.0	...	1.0	1.0	1.0	1.0	48.9
000edc09-fcfa	1976	0.0	0.0	...	0.0	0.0	0.0	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

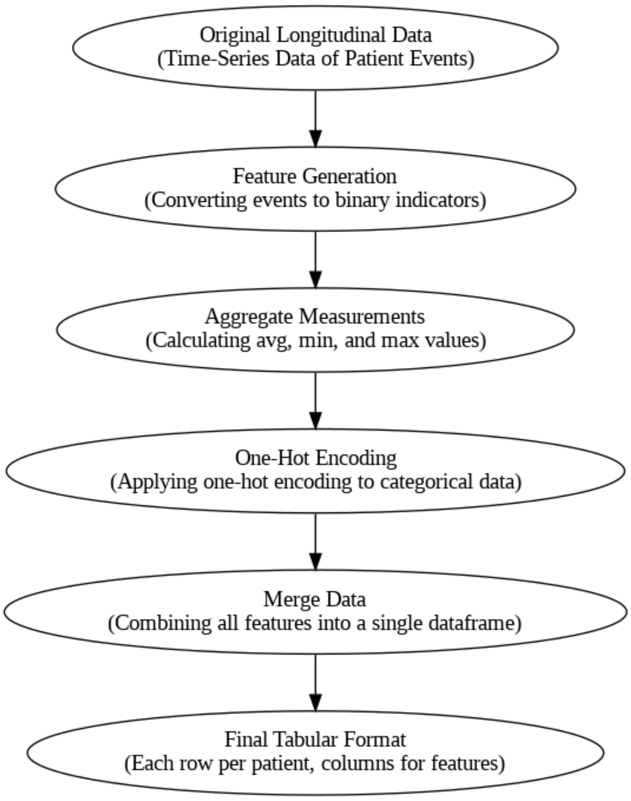

Fig. 1: Preprocessing Pipeline - Converting Longitudinal Data to Tabular Data

3.2.2 Time Series Dataset with LSTM

In This method, we aim to leverage the temporal characteristics of the data through a Long Short-Term Memory (LSTM) network. The process involves converting each patient's sequential data into a series of embedded vectors, allowing the LSTM to process and learn from the evolving patient health data over time.

Methodology for Data Preparation

– **Sequence Construction:** For each patient, we process their data from individual files, constructing sequences of temporal features from the patient events. Each feature in the sequence represents a combination of vectors of the `rela_name` (Table 3a) and `tail_type:tail_name` (Table 3b) components, embedded using the corresponding pre-trained vectors (Table 3). These sequences are labeled based on the presence or absence of the `Injury of Heart` condition.

– **Data Splitting:** The data is then split into training, validation, and test sets based on predefined patient IDs, ensuring consistency across all methods. We scale the features using a StandardScaler to normalize the data, which improves the convergence of the LSTM during training.

The input data for the model is a three-dimensional time-series dataset. Each patient's data is represented as a sequence of temporal features embedded into vectors, allowing the model to process and learn from the evolving patient health data over time.

Model Training and Evaluation

– **Building the Model:** The LSTM model is constructed with two bidirectional LSTM layers followed by a Dense layer. The bidirectional LSTM layers allow the model to learn from both past and future contexts of the sequence, enhancing the model's understanding of the temporal dynamics (Figure 2).

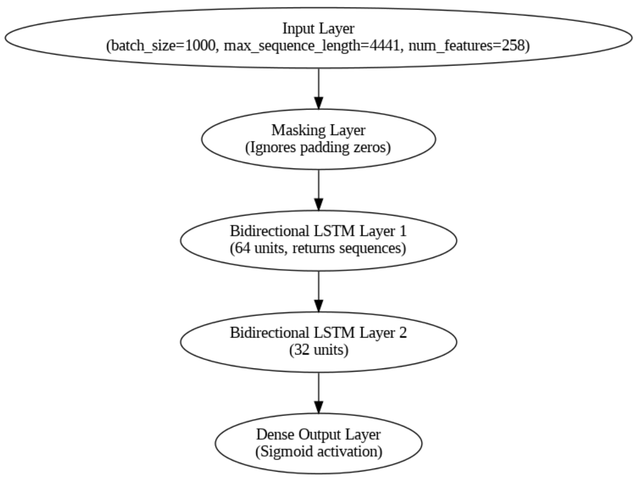

Fig. 2: LSTM Model Architecture

– **Training and Evaluation:** The model is trained using binary cross-entropy loss and optimized with the Adam optimizer. We evaluate the model using

Table 3: Embedding Pre-Trained Vectors

(a) Relation Data

relation	0	1	2	3	...
rBirthYear	-0.147	-0.511	0.092	0.303	...
rCareplan	0.147	-0.671	0.090	0.301	...
rCareplanReason	-0.254	-0.577	0.022	-0.183	...
rCondition	-0.181	-0.856	0.070	0.452	...
⋮	⋮	⋮	⋮	⋮	⋮

(b) Tail Data

tail	0	1	2	3	...
BirthYear:1958	0.182	-0.071	-0.055	-0.200	...
Careplan:698360004	-0.113	0.095	-0.056	-0.212	...
Careplan:736285004	-0.101	0.094	-0.066	-0.243	...
Careplan:736376001	-0.114	0.093	-0.055	-0.222	...
⋮	⋮	⋮	⋮	⋮	⋮

an F1 score metric to balance the trade-off between precision and recall, which is crucial for the imbalanced nature of our binary classification task. The training proceeds in steps, with regular evaluation of the validation set to determine the optimal threshold for binary classification based on predicted probabilities.

– **Threshold Optimization:** At each training step, we utilize precision-recall curves to find the optimal threshold for converting prediction probabilities into binary predictions. This threshold is chosen to maximize the F1 score on the validation set, ensuring the best balance between false positives and false negatives.

– **Model Performance:** The final trained model, selected from the epoch with the highest F1 score on the validation set, is evaluated on the test set to assess its predictive performance. The evaluation includes calculating the F1 score and generating a confusion matrix to understand the model's classification capabilities.

3.2.3 Patient Window Breakdown

This method aims to leverage staggered time windows to capture a patient's health information over time. Each window represents a fixed-duration segment of patient events, allowing for more granular temporal prediction. Multiple records for each patient are created based on staggered time windows to capture detailed health information over time. Each window's data is first processed using XGBoost to predict probabilities of health outcomes, which are then used as inputs to an LSTM model, forming a sequence-to-sequence prediction framework.

Methodology for Data Preparation

Each patient's record is transformed into a series of windows, each representing a fixed duration of 1000 days. This windowing method helps structure the data by summarizing the health events occurring within each period, thereby maintaining the essential temporal dynamics while allowing for study using 2D data formats. We pre-process the records in a manner similar to the one-hot encoding process in the first method, converting event descriptions into binary indicators of occurrence within each window, including demographics and health event occurrences (Figure 3).

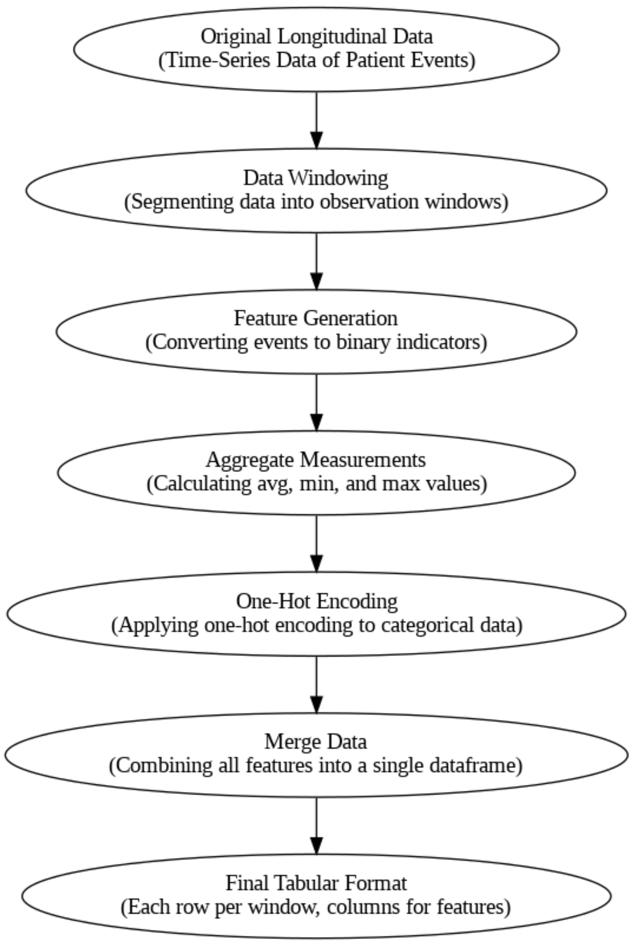

Fig. 3: Preprocessing Pipeline - Converting Longitudinal Data to Tabular Data (Multiple Time Windows)

– **Window Construction:** For each patient, we divide their health records into time windows of

1000 days. We construct these windows by extracting the relevant events within each time window and including demographic information (Table 4).

– **One-Hot Encoding:** We transform the health events into one-hot encoded features for each time window. For categorical features, a binary indicator of occurrence is used, while for numerical features, we calculate the average, minimum, and maximum values within the window.

– **Data Splitting:** The resulting data is split into training, validation, and test sets based on predefined patient IDs, ensuring consistency across all methods and facilitating consistent evaluation across different modeling techniques.

– **Label Alignment:** We align the patient labels with the windowed data, replicating the label for each patient's corresponding windows. This ensures that each window receives the same label as the patient's overall diagnosis.

The input data for the model consists of staggered time windows for each patient. Each window is initially represented in a two-dimensional tabular format for XGBoost processing and then converted into sequences for the LSTM model. This hybrid method captures detailed temporal dynamics and facilitates sequential prediction.

Model Training and Evaluation (Figure 4)

– **Building the XGBoost Model:** The XGBoost model is trained on the windowed data to generate prediction probabilities for each window. We optimize the maximum tree depth between 5 and 30 to achieve the best F1 score on the tuning set. The best model is used to generate prediction probabilities for the training, validation, and test sets.

– **Combining Predictions and Additional Features:** After the XGBoost training, we add demographic data, including `gender`, `race`, `ethnicity`, `maritalstatus`, `county`, and statistical data such as the number of medications, allergies, conditions, procedures, care plans, and immunizations for each time window, alongside the predicted probabilities from the XGBoost model. The final features, including the predictions and these additional attributes, are then used as input for the LSTM model.

– **Building the LSTM Model:** The LSTM model utilizes the prediction probabilities from the XGBoost model as input features, along with other relevant features from the windowed data. The model consists of two LSTM layers followed by a Dense layer, enabling sequential prediction of the windowed data (Figure 5).

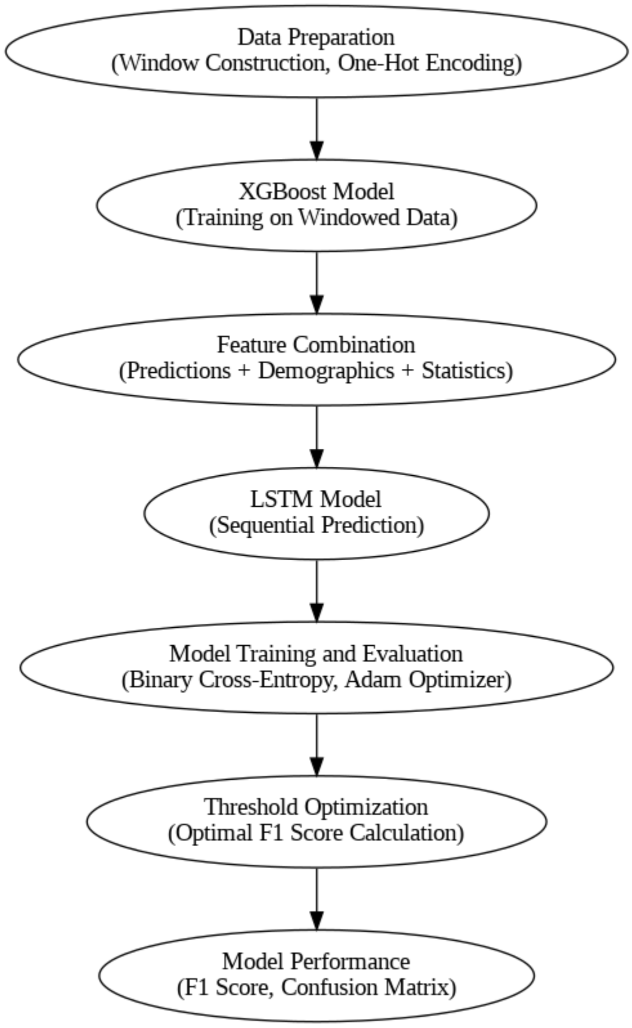

Fig. 4: Hybrid Predictive Model Workflow Combining XGBoost and LSTM

– **Training and Evaluation:** The LSTM model is trained using binary cross-entropy loss and optimized with the Adam optimizer. We evaluate the model using an F1 score metric to balance precision and recall, which is crucial for the imbalanced nature of our binary classification task. The training proceeds in steps, with regular evaluation of the validation set to determine the optimal threshold for binary classification. The models are evaluated based on their F1 score and computational efficiency. The F1 score, a harmonic mean of precision and recall, is particularly useful in the imbalanced datasets typical of medical event predictions. Computational efficiency is assessed by the time required to train and predict using each model configuration.

– **Threshold Optimization:** At each training step, we utilize precision-recall curves to find the optimal threshold for converting prediction probabili-

Table 4: Example of All Patient Records in One Table after Dividing into Windows

patient	birth_yr	days_back	days_fore	gender_male	...	condition_230690007	medication_1000126	observation_qols_min	observation_qaly_min
0abbf153-7241_window_12	1969	7228.0	11396.0	0.0	...	0.0	0.0	-	48.9
0abbf153-7241_window_17	1969	2272.0	16352.0	0.0	...	0.0	1.0	0.0	-
0abbf153-7241_window_19	1969	0.0	18624.0	0.0	...	1.0	0.0	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
00051dc6-38dc_window_8	1998	0.0	7824.0	0.0	...	-	0.0	0.0	-
000edc09-fcfa_window_10	1976	5840.0	9968.0	0.0	...	-	0.0	0.0	-
000edc09-fcfa_window_11	1976	5098.0	10710.0	0.0	...	-	0.0	0.0	-
ffaa918c-d035_window_7	1945	20597.0	6663.0	0.0	...	-	0.0	0.0	-
ffb28c11-110c_window_10	1985	2732.0	9996.0	1.0	...	-	0.0	0.0	-
ffb28c11-110c_window_13	1985	0.0	12728.0	1.0	...	-	0.0	0.0	-

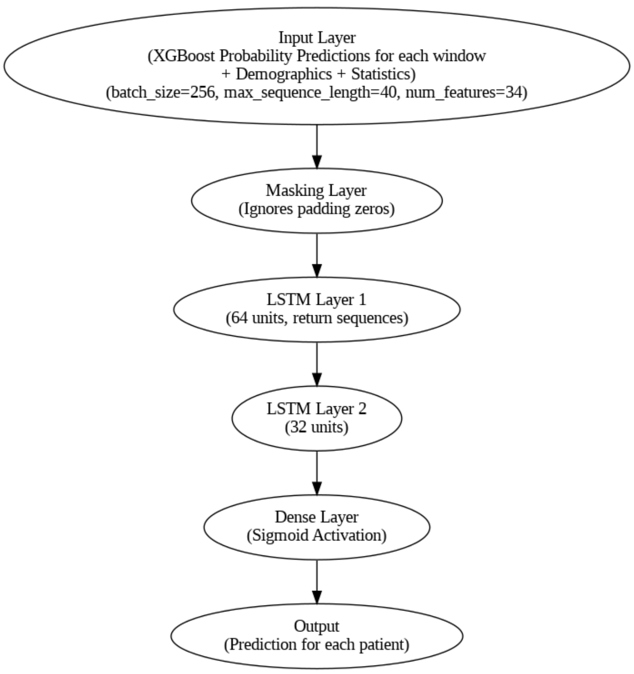

Fig. 5: Detailed LSTM Model Architecture

ties into binary predictions. This threshold is chosen to maximize the F1 score on the validation set, ensuring the best balance between false positives and false negatives.

– **Model Performance:** The final trained model is evaluated on the test set to assess its predictive performance. The evaluation includes calculating the F1 score and generating a confusion matrix to understand the model's classification capabilities.

4 Results and Discussion

In this section, we present a detailed analysis of the performance of three different machine learning methods for predicting health outcomes from patient records.

The methods evaluated are:

1. **XGBoost with Tabular Dataset**: This method uses a 2D tabular format of the dataset.
2. **LSTM with Time Series Dataset**: This method leverages the temporal characteristics of the data through a Long Short-Term Memory (LSTM) network, converting each patient's sequential data into a series of embedded vectors.
3. **Hybrid XGBoost then LSTM**: This method combines the strengths of XGBoost and LSTM, using XGBoost to generate prediction probabilities for each window of data, which are then used as inputs for an LSTM model.

4.1 Comparative Analysis

Table 5 provides a summary of the performance metrics for each method, including the F1 score, training time, best F1 score achieved, the optimal threshold for F1 score, and the F1 score if the data were balanced. Figure 6 visually depicts the F1 scores and run times for each method, supporting the comparison results presented in Table 5.

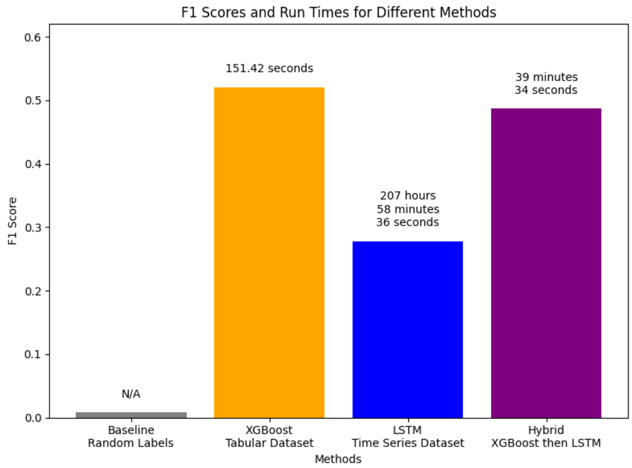

Fig. 6: F1 Scores and Run Times for Different Methods

In evaluating the performance of our models, we introduced a baseline using random labels. This baseline serves as a control to demonstrate the performance improvement of our models over random predictions. The baseline model randomly assigns labels to the data, providing a point of reference for assessing the effectiveness of our machine-learning approaches. By comparing the F1 scores of our models to the baseline, we can quantitatively measure the extent to which our models outperform random guessing, thereby validating their predictive capabilities. The F1 score of the baseline model is 0.0081, which is substantially lower than the F1 scores of the XGBoost, LSTM, and hybrid methods. This comparison clearly demonstrates that our models are not only functioning better than random chance but also providing meaningful and accurate predictions.

The XGBoost model shows the highest F1 score with the shortest training time, highlighting its efficiency and effectiveness. In contrast, the LSTM model, despite capturing temporal dependencies, has a significantly lower F1 score and the longest training time, indicating high computational costs and overfitting issues. The hybrid method improves over the standalone LSTM model but still falls short of XGBoost's performance, both in terms of accuracy and computational efficiency.

4.2 Method 1: XGBoost with Tabular Dataset

The XGBoost model demonstrates the highest F1 score of 0.5199 and requires relatively low computational resources, with a training time of 151.42 seconds. The optimal tree depth is 19, achieving the best F1 score of 0.5542, as depicted in Figure 7.

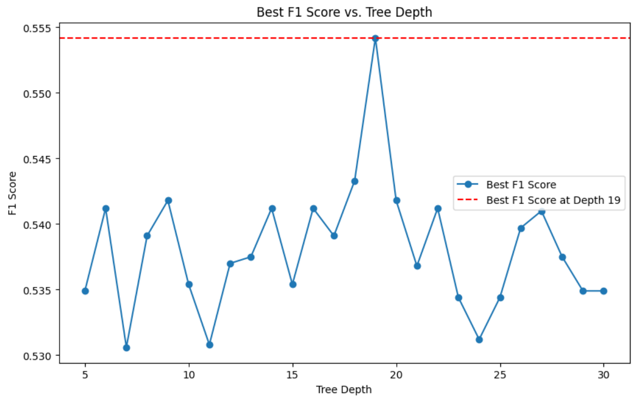

Fig. 7: Method 1 - XGBoost Model Best F1 Score vs. Tree Depth

XGBoost excels in this context due to several key reasons:

– **Efficiency in Handling Large Datasets**: XGBoost is designed to handle large-scale datasets efficiently, making it well-suited for the extensive patient records in this study [6]. The model's ability to parallelize tree construction and optimize memory usage contributes to its fast training time.

– **Robustness Against Overfitting**: The ensemble nature of gradient boosting allows XGBoost to capture complex patterns in the data while maintaining robustness against overfitting. The model's regularization techniques, including L1 and L2 regularization, help prevent overfitting, which is crucial for generalizing well on unseen data [17]. These techniques help manage model complexity and improve performance on large datasets by penalizing more complex models, thereby enhancing their predictive capabilities and stability.

– **Effective Utilization of Categorical Features**: One-hot encoding transforms categorical features into binary indicators, enabling XGBoost to effectively utilize the diverse patient information. This approach ensures that all relevant features are incorporated into the model (even though it cannot capture the temporal characteristic of the data), enhancing its predictive power.

– **Feature Importance and Interpretability**: XGBoost provides insights into feature importance, allowing for better interpretability of the model [6]. Understanding which features contribute most to the predictions is valuable in the healthcare con-

Table 5: Comparison of Different Methods

Method	F1 Score	Training Time *(seconds)*	Best F1 Score	F1 Optimal Threshold	F1 Score if Data is Balanced
Baseline Random Labels	0.0081	–	–	–	0.0159
XGBoost Tabular Dataset	0.5199	151.42	0.5542	0.0113	0.9415
LSTM Time Series Dataset	0.2781	748716.05	0.3017	0.0605	0.3622
Hybrid XGBoost then LSTM	0.4864	2374.59	0.5068	0.1850	0.8423

text, where interpretability is often as important as accuracy.

4.3 Method 2: LSTM with Time Series Dataset

The LSTM model, designed to leverage the temporal characteristics of the data, achieves an F1 score of 0.2781 with a best F1 score of 0.3017. However, the training time is significantly higher, amounting to 748716.05 seconds (approximately 8.66 days). The performance trend is shown in Figure 8.

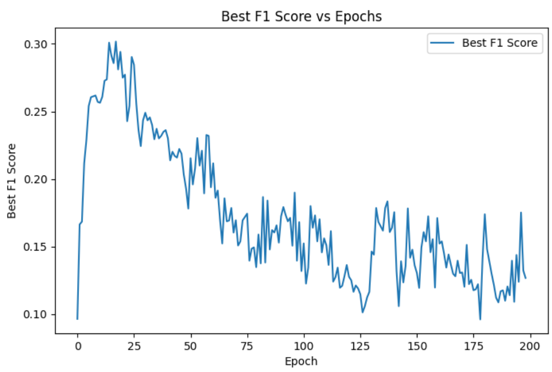

Fig. 8: Method 2 - LSTM Model Best F1 Score vs. Epochs

The LSTM model's poor performance can be attributed to several factors:

– **High Computational Cost**: LSTM models are inherently complex and computationally intensive. Training such models requires significant resources and time, which is evident in the exceptionally long training time observed in this study.

– **Sensitivity to Data Quality**: LSTM's reliance on sequential data processing makes it sensitive to the quality and continuity of the time series data. Any irregularities or noise in the data can significantly impact the model's performance. The synthetic nature of the dataset and potential inconsistencies in the time series may hinder the LSTM's ability to learn effective patterns.

– **Difficulty in Achieving Stable Performance**: The high variance observed in the F1 score over epochs indicates challenges in achieving stable and consistent performance. LSTM models often require careful tuning of hyperparameters and a large amount of high-quality data to perform well, which may not be fully met in this study.

– **Complexity of Capturing Long-Term Dependencies**: While LSTM models are designed to capture long-term dependencies, doing so effectively requires a large and representative dataset. The complexity of the model and the high dimensionality of the data can lead to difficulties in learning meaningful long-term patterns.

– **Possibility of Overfitting**: The training and validation loss curves for the LSTM model, shown in Figure 9, provide further insight into the model's performance. The training loss decreases steadily, indicating that the model is learning and fitting well with the training data. However, the validation loss increases, suggesting that the model is starting to overfit and indicating that the model is no longer improving on the validation data and is instead becoming overly specialized to the training data. Overfitting occurs when the model captures noise and details specific to the training data, which negatively impacts its performance on new, unseen data. This is evident from the widening gap between the training and validation loss, highlighting the model's reduced generalization capability. This trend underscores the challenges of using LSTM models with high-dimensional and potentially noisy datasets, as they can easily overfit without careful regularization and sufficient data quality.

4.4 Method 3: Hybrid XGBoost then LSTM

The hybrid method achieves an F1 score of 0.4864, with a best F1 score of 0.5068 and a training time of 2374.59 seconds. This method aims to combine the strengths of both XGBoost and LSTM. Figure 10 demonstrates that the F1 score improves rapidly during the initial epochs and stabilizes, maintaining a high level of per-

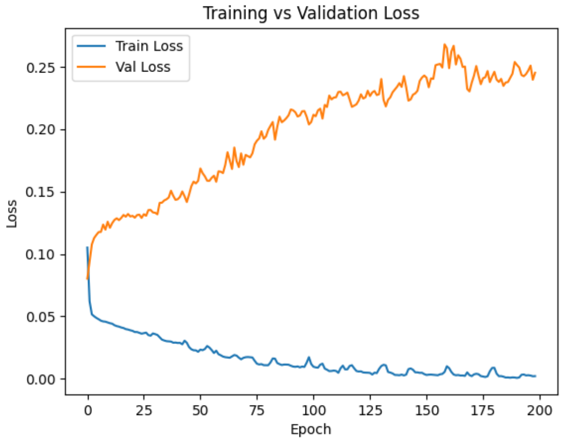

Fig. 9: Method 2 - LSTM Model Training vs Validation Loss

formance throughout the remaining epochs. This indicates that the hybrid method quickly learns to leverage the strengths of both models and then consistently applies this knowledge. The consistency in performance throughout the remaining epochs highlights the model's ability to generalize well to new data, reducing the likelihood of overfitting, which is a problem in Method 2 mentioned above.

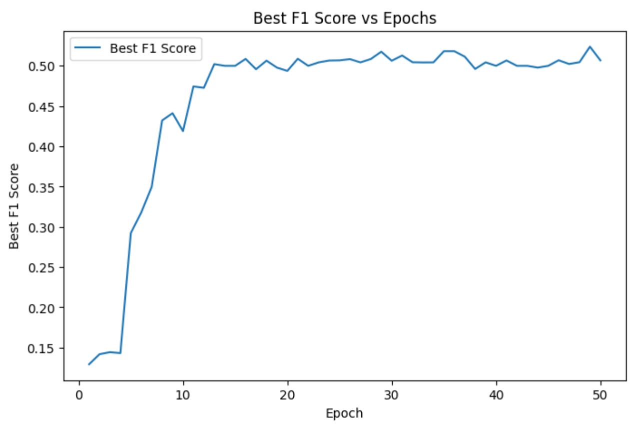

Fig. 10: Method 3 - LSTM Model of Hybrid Method Best F1 Score vs. Epoch

The hybrid method, while improving performance compared to using LSTM alone, does not surpass the efficiency and accuracy achieved by the XGBoost model with the tabular dataset. The potential reasons for this are multifaceted:

- **Integration Challenges**: Combining two different models introduces additional complexity. Although XGBoost effectively captures static patterns in the tabular data, the subsequent LSTM model struggles to leverage the temporal dynamics of the already processed data. The integration of prediction probabilities as inputs for the LSTM may lead to information loss or redundancy. Specifically, the predictions from XGBoost are aggregated results, and while they provide a summary of static features, they may not carry forward the intricate temporal relationships necessary for the LSTM to function optimally. This can result in a loss of detailed temporal information that the LSTM needs to make accurate predictions, thereby hindering its performance.
- **Overhead of Model Combination**: The hybrid method involves multiple stages of data processing and model training, leading to increased overhead. This added complexity does not translate into a significant performance gain, as the individual advantages of XGBoost and LSTM are not fully synergized. Each stage of the hybrid model requires separate tuning, data preprocessing, and validation, which collectively increases the computational burden. Moreover, the intermediate data transformations and model handovers can introduce inefficiencies, further complicating the model training process and negating potential performance benefits.
- **Balancing Static and Temporal Information**: The hybrid method aims to balance the strengths of capturing static features with XGBoost and temporal dependencies with LSTM. However, the difficulty in harmonizing these two aspects may limit the overall effectiveness. The temporal dynamics captured by the LSTM may be overshadowed by the static predictions from XGBoost, resulting in suboptimal performance. Essentially, while XGBoost excels at capturing feature interactions in a static context, it may dominate the hybrid model's predictions, leaving the temporal nuances that the LSTM could capture underutilized. This imbalance can prevent the hybrid model from fully capitalizing on the temporal relationships in the data, leading to performance that is less than optimal compared to the pure XGBoost model.
- **Computational Efficiency**: While the hybrid method is more computationally efficient than the standalone LSTM model, it still requires more resources than the XGBoost model alone. The added computational cost does not yield proportional improvements in predictive performance.

The comparative analysis of XGBoost, LSTM, and the hybrid method underscores the efficiency and accuracy of XGBoost when applied to a tabular dataset for predicting health outcomes from patient records. Despite the theoretical advantages of leveraging LSTM for capturing temporal dependencies, the practical challenges of high computational costs, sensitivity to data quality, and the risk of overfitting limit its effectiveness in this study. The hybrid approach, while mitigating some of the drawbacks of standalone LSTM, does not achieve the same level of performance as XGBoost due to integration complexities and computational overhead. Consequently, XGBoost remains the preferred method in this context, offering a balanced solution that maximizes predictive accuracy and computational efficiency for longitudinal EHR data.

5 Conclusions

In this study, we evaluated three machine learning approaches for predicting health outcomes from longitudinal patient records: (1) XGBoost with a tabular dataset, (2) LSTM with a time series dataset, and (3) a hybrid XGBoost then LSTM approach. Our results indicate that the XGBoost model with a tabular dataset is the most effective approach, achieving an F1 score of 0.5199 with relatively low computational resources. This method demonstrated efficiency in handling large-scale datasets, robustness against overfitting, effective utilization of features, and better interpretability through feature importance analysis.

The LSTM model, while designed to capture temporal dependencies, showed a lower F1 score of 0.2781 and suffered from high computational costs, amounting to approximately 8.66 days of training time. The training and validation loss curves revealed significant overfitting, indicating challenges in achieving stable and consistent performance with high-dimensional and potentially noisy datasets.

The hybrid approach, combining XGBoost and LSTM, achieved an F1 score of 0.4864. Although it improved performance compared to using LSTM alone, it did not surpass the efficiency and accuracy of the standalone XGBoost model. The integration of the two models introduced additional complexity and overhead, limiting the overall effectiveness of the hybrid approach.

Future work should focus on optimizing the hybrid approach to better leverage the strengths of XGBoost and the inherent temporal characteristics of the dataset. This could involve developing advanced methods for integrating temporal dynamics into XGBoost predictions or exploring other model architectures that can effectively combine static and temporal features. Specifically, enhancing the interaction between XGBoost and LSTM by developing more sophisticated data transformation techniques that preserve temporal information could lead to better performance. Additionally, investigating ways to directly incorporate temporal dependencies within XGBoost itself might streamline the hybrid approach.

Advanced preprocessing techniques should also be a key area of focus. Techniques such as automated feature selection, which uses algorithms to select the most relevant features automatically, and dimensionality reduction, which reduces the number of variables under consideration, can significantly improve model performance. These techniques can help in managing the high dimensionality and complexity of longitudinal health data, while still keeping as much data as possible. Moreover, incorporating domain-specific knowledge into preprocessing can improve data quality. For instance, using medical ontologies to refine feature selection could lead to more clinically relevant models.

Exploring alternative integration strategies, like combining XGBoost features with raw inputs before feeding them into the LSTM, may better capture feature interactions and temporal dependencies. This could involve creating a hybrid architecture where the outputs of XGBoost are used as additional features rather than the sole inputs to the LSTM, allowing the LSTM to leverage both the processed outputs and the raw temporal data.

Moreover, attention mechanisms and transformer models present promising avenues for future research. Attention mechanisms allow models to focus on important parts of the input sequence, improving the capture of relevant temporal dynamics. Transformers, which use self-attention mechanisms to weigh the importance of different parts of the input data, can be particularly effective in handling sequential data and capturing long-range dependencies. These techniques could provide more sophisticated ways to integrate temporal information, maintain computational efficiency, and improve predictive performance. Extensive hyperparameter tuning using automated optimization techniques, such as Bayesian optimization or grid search, can also be used to find the best configurations for both XGBoost and LSTM, ensuring optimal performance.

Evaluating these approaches on larger and more diverse datasets will be crucial to generalizing the findings and ensuring robustness across different healthcare applications. This could involve using real-world clinical datasets from various healthcare institutions to validate the models' effectiveness and adaptability. Overall, the results of this study highlight the potential of machine learning techniques to improve health outcome predictions from patient records. By addressing the identified limitations and exploring new methodologies, future research can continue to advance the field and contribute to more accurate and efficient healthcare solutions.

References

1. Aguiar, H., Santos, M., Watkinson, P., Zhu, T.: Learning of cluster-based feature importance for electronic health record time-series. Machine Learning Research **17**, 45–58 (2022). DOI 10.1007/s10994-022-05981-2. URL https://link.springer.com/article/10.1007/s10994-022-05981-2
2. Alam, A.N.M.S., Reza, R., Abrar, A., Ahmed, T., Ahmed, S., Sharar, S., Rasel, A.A.: Patients' severity states classification based on electronic health record (ehr) data using multiple machine learning and deep learning approaches. IEEE Access **9**, 130111–130125 (2021). DOI 10.1109/ACCESS.2021.3119812. URL https://ieeexplore.ieee.org/document/9572071
3. Beeksma, M., Verberne, S., van den Bosch, A., Das, E., Hendrickx, I., Groenewoud, S.: Predicting life expectancy with a long short-term memory recurrent neural network using electronic medical records. BMC Medical Informatics and Decision Making **19**(1), 77 (2019). DOI 10.1186/s12911-019-0775-2. URL https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0775-2
4. Brisimi, T.S., Xu, T., Wang, T., Dai, W., Adams, W., Paschalidis, I.: Predicting chronic disease hospitalizations from electronic health records: An interpretable classification approach. Proceedings of the IEEE **106**, 690–707 (2018). DOI 10.1109/JPROC.2017.2789319. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6419763/
5. Cava, W.L., Bauer, C., Moore, J.H., Pendergrass, S.A.: Interpretation of machine learning predictions for patient outcomes in electronic health records. In: AMIA Annual Symposium Proceedings, pp. 572–581 (2019). URL https://pubmed.ncbi.nlm.nih.gov/32308851
6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. arXiv preprint arXiv:1603.02754 (2016). URL https://ar5iv.labs.arxiv.org/html/1603.02754
7. Choi, E., Schuetz, A., Stewart, W.F., Sun, J.: Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association **24**, 361–370 (2017). DOI 10.1093/jamia/ocw112. URL https://doi.org/10.1093/jamia/ocw112
8. Christensen, T., Frandsen, A., Glazier, S., Humpherys, J., Kartchner, D.: Machine learning methods for disease prediction with claims data. IEEE pp. 467–4674 (2018). DOI 10.1109/ICHI.2018.00108. URL https://ieeexplore.ieee.org/document/8419439
9. Feng, A.: Accurate covid-19 health outcome prediction and risk factors identification through an innovative machine learning framework using longitudinal electronic health records. IEEE Journal of Biomedical and Health Informatics **25**(8), 2941–2951 (2021). DOI 10.1109/JBHI.2021.3096234. URL https://ieeexplore.ieee.org/document/9465708
10. Huang, Y., Yang, X., Xu, C.: Time-guided high-order attention model of longitudinal heterogeneous healthcare data. Springer (2019). DOI 10.1007/978-3-030-29908-8_5. URL https://link.springer.com/chapter/10.1007/978-3-030-29908-8_5
11. Jin, L.P., Dong, J.: Ensemble deep learning for biomedical time series classification. Computational Intelligence and Neuroscience (2016). DOI 10.1155/2016/6212684. URL https://doi.org/10.1155/2016/6212684
12. Johnson, A.E.W., Pollard, T.J., Mark, R.G.: Machine learning and decision support in critical care. Proceedings of the IEEE **104**(2), 444–466 (2016). DOI 10.1109/JPROC.2015.2501978. URL https://ieeexplore.ieee.org/document/7372035
13. Liu, J., Yao, J., Zhou, Q., Wang, Z., Huang, L.: Lstmaedwsslm: A unified approach for imbalanced time series data classification. Applied Intelligence **53**, 21077–21091 (2023). DOI 10.1007/s10489-023-04642-0. URL https://doi.org/10.1007/s10489-023-04642-0
14. Liu, M., Guo, C., Guo, S.: An explainable knowledge distillation method with xgboost for icu mortality prediction. Journal of Biomedical Informatics **122**, 103878 (2022). DOI 10.1016/j.jbi.2021.103878. URL https://www.sciencedirect.com/science/article/pii/S1532046421002935
15. Liu, S., Fu, B., Wang, W., Liu, M., Sun, X.: Dynamic sepsis prediction for intensive care unit patients using xgboost-based model with novel time-dependent features. IEEE Journal of Biomedical and Health Informatics **26**, 4258–4269 (2022). DOI 10.1109/JBHI.2022.3171673. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9767649
16. Moskovitch, R., Polubriaginof, F.C.G., Weiss, A., Ryan, P., Tatonetti, N.: Procedure prediction from symbolic electronic health records via time intervals analytics. Journal of biomedical informatics **75**, 70–82 (2017). DOI 10.1016/j.jbi.2017.07.018. URL https://pubmed.ncbi.nlm.nih.gov/28823923/
17. Mukherjee, S.: Xgboost: A comprehensive guide, model overview, analysis, and code demo using paperspace gpus. Paperspace (2020). URL https://blog.paperspace.com/xgboost-a-comprehensive-guide/
18. Nguyen, D.K., Lan, C.H., Chan, C.L.: Deep ensemble learning approaches in healthcare to enhance the prediction and diagnosing performance: The workflows, deployments, and surveys on the statistical, image-based, and sequential datasets. International Journal of Environmental Research and Public Health **18** (2021). DOI 10.3390/ijerph182010811. URL https://doi.org/10.3390/ijerph182010811
19. Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Liu, P.J., Liu, X., Sun, M., Sundberg, P., Yee, H., Zhang, K., Duggan, G.E., Flores, G., Hardt, M., Irvine, J., Le, Q., Litsch, K., Marcus, J., Mossin, A., Tansuwan, J., Wang, D., Wexler, J., Wilson, J., Ludwig, D., Volchenboum, S.L., Chou, K., Pearson, M., Madabushi, S., Shah, N.H., Butte, A.J., Howell, M.D., Cui, C., Corrado, G.S., Dean, J.: Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine **1**(18) (2018). DOI 10.1038/s41746-018-0029-1. URL https://www.nature.com/articles/s41746-018-0029-1
20. Rajliwall, N.S., Davey, R., Chetty, G.: Cardiovascular risk prediction based on xgboost. In: 2018 5th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 246–252 (2018). DOI 10.1109/APWConCSE.2018.00047. URL https://ieeexplore.ieee.org/document/8853798
21. Shickel, B., Tighe, P.J., Bihorac, A., Rashidi, P.: Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. IEEE Journal of Biomedical and Health Informatics **22**(5), 1589–1604 (2018). DOI 10.1109/JBHI.2017.2767063. URL https://ieeexplore.ieee.org/document/8051139

22. Sofrygin, O., Zhu, Z., Schmittdiel, J.A., Adams, A.S., Grant, R.W., van der Laan, M.J., Neugebauer, R.: Using machine learning to analyze electronic health record data: a novel approach to data curation. Journal of the American Medical Informatics Association **28**(10), 2065–2075 (2021). DOI 10.1093/jamia/ocab100. URL `https://academic.oup.com/jamia/article/28/10/2065/6306216`
23. Tsiklidis, E.J., Sinno, T., Diamond, S.L.: Predicting risk for trauma patients using static and dynamic information from the mimic iii database. PLOS ONE **17**, 1–13 (2022). DOI 10.1371/journal.pone.0262523. URL `https://doi.org/10.1371/journal.pone.0262523`
24. Ugale, P.A., Gadakh, A., Malunjkar, A., Sawant, R., Dhakane, V.: Integrated healthcare system using machine learning. International Journal for Research in Applied Science and Engineering Technology (2022). DOI 10.22214/ijraset.2022.47520. URL `https://www.ijraset.com/best-journal/integrated-healthcare-system-using-machine-learning-110`
25. Walonoski, J., Klaus, S., Granger, E., Hall, D., Gregorowicz, A., Neyarapally, G., Watson, A., Eastman, J.: Synthea™ novel coronavirus (covid-19) model and synthetic data set. Informatics in Medicine Unlocked **20**, 100007 (2020). DOI 10.1016/j.ibmed.2020.100007. URL `https://doi.org/10.1016/j.ibmed.2020.100007`. Licensed under CC BY 4.0