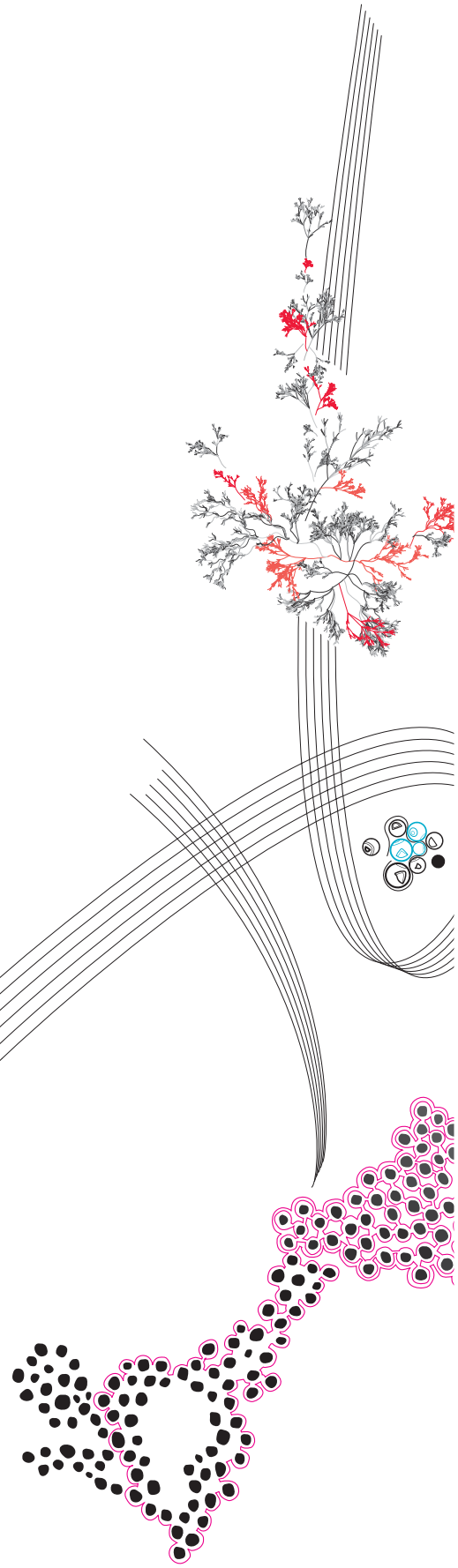


MSc Computer Science
Final Project

From Laboratory Test Results to Emergency Department Admission Status: Forecasting with Machine Learning and Predictive Process Mining

Priya Naguine

Supervisor: Faiza Bukhsh, Hans Krabbe & Duc Viet Le

August, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

Emergency department overcrowding is a significant issue impacting healthcare systems globally, influencing patient care and resource allocation. This study investigates whether predictive process mining offers an improvement over traditional machine learning methods for classifying emergency department admissions using sequential medical data. By leveraging the MIMIC-IV dataset, which includes laboratory tests conducted during hospital admissions and captures dynamic changes in test results over time, the research compares the performance of predictive process mining and machine learning models. Results show that the standalone machine learning model, which differs from predictive process mining models primarily in the data itself and its format, has a performance comparable to the predictive process mining model with event-level features. However, it outperforms the predictive process mining model with case-level features in terms of accuracy, precision and recall. The study also identifies limitations, such as the exclusion of general practitioner visits and pre-hospitalisation tests from the dataset and challenges related to class imbalance, which impact model training and generalisability.

Keywords: classification, deep learning, emergency department, machine learning, medical laboratory tests, predictive process mining, time-series analysis

Chapter 1

Introduction

The healthcare industry is undergoing a transformative shift towards data-driven decision-making, driven by technological advancements and the increasing availability of electronic health records (EHRs). In this evolving landscape, machine learning (ML) and process mining (PM) have emerged as powerful tools for enhancing patient care and operational efficiency [36, 49].

ML involves training algorithms to recognise patterns and make decisions with minimal human intervention [1]. It has shown effectiveness in disease diagnosis, risk assessment and outcome prediction [2, 26, 37, 41]. PM, on the other hand, focuses on analysing business processes based on event logs extracted from information systems [44]. This approach provides insights into process flows and deviations, helping to optimise operations, especially in complex environments like healthcare [7, 27, 34].

Predictive process mining (PPM) extends traditional PM into the realm of predictive analytics by applying data mining and ML techniques to forecast future process states based on historical data [9]. PPM offers a detailed view of patient histories and anticipates future events, enabling proactive decision-making.

This study compares PPM with traditional ML methods for predicting emergency department (ED) admissions using medical laboratory test data. The primary aim is to forecast whether a patient will be admitted to the ED based on a six-month history of laboratory tests. Both ML and PPM methods utilise deep learning models to predict ED admissions, but they differ primarily in the data itself and its format. This difference in how data is processed influences each approach's ability to leverage information for enhancing prediction accuracy.

Predicting ED admissions is crucial for healthcare facilities as it enables early intervention and effective resource allocation. Accurate predictions allow healthcare providers to anticipate patient demand surges, facilitating proactive adjustments in staffing, resources and treatment protocols [5]. This not only enhances patient care by reducing waiting times and overcrowding but also improves operational efficiency by ensuring that resources are allocated adequately. Furthermore, predictive modelling supports strategic healthcare management by forecasting patient surges, which helps minimise ED overcrowding risks and associated compromises in patient care and costs [4]. Ultimately, this proactive approach enhances emergency responsiveness, optimises workflows and improves overall patient outcomes.

In this context, the study aims to determine whether PPM significantly outperforms traditional ML methods in predicting ED admissions. To achieve this, it leverages a comprehensive dataset of medical laboratory test results spanning six months. The data is pre-processed and the models are evaluated to compare the effectiveness of both ML and

PPM approaches.

The main research question guiding this study, along with three sub-research questions, is as follows:

To what extent can predictive process mining improve the classification of emergency department admissions using medical laboratory test data compared to standalone machine learning methods?

1. **What are the specific data requirements and format differences between predictive process mining techniques and standalone machine learning methods?**
2. **How do the learning curves of predictive process mining techniques compare to those of the standalone machine learning model?**
3. **How does the predictive performance of predictive process mining techniques compare to standalone machine learning models?**

The remainder of this paper is structured as follows: Section 2 gives the background information required for this research. Section 3 outlines the methodology adopted, while Section 4 provides an in-depth analysis of it. Section 5 presents and discusses the findings from the analysis. Section 6 explores the broader implications and significance of these results. Finally, Section 7 summarises the research outcomes and suggests directions for future research.

Chapter 2

Background

This chapter provides a comprehensive overview of the foundational concepts relevant to this research. It begins with an examination of neural networks in Section 2.1, including deep learning in Section 2.1.1 and specific architectures such as recurrent neural networks, long short-term memory networks and gated recurrent units in Section 2.1.2. The principles of process mining are then introduced in Section 2.2, leading to a discussion on predictive process mining techniques in Section 2.2.1. This background information establishes the foundation for the comparative analysis presented in this study.

2.1 Neural networks (NNs)

NNs are computational models inspired by early theories on how the human brain processes information [24]. They consist of interconnected layers of neurons working together to solve specific problems. Typically, a NN comprises three types of layers: the input layer, the hidden layer(s) and the output layer.

The input layer receives raw data, which is then processed by the hidden layer. The hidden layer extracts and learns patterns from the input data. Finally, the output layer generates the network's prediction or classification based on the patterns learned in the hidden layer.

Each connection between neurons in a NN is assigned a weight, determining the strength of influence one neuron has on another. Each neuron applies an activation function to its inputs, transforming them into outputs that contribute to the network's overall prediction or classification.

NN models offer several advantages that make them appealing for various applications [42]. One significant advantage is that they require less formal statistical training compared to traditional statistical models. NNs can implicitly detect complex non-linear relationships between independent and dependent variables, capturing interactions that might be missed by other models. They can identify all possible interactions between predictor variables, providing a comprehensive understanding of the data. Additionally, NNs offer flexibility in training, as they can be developed using various training algorithms.

However, NNs also have notable disadvantages. They often operate as a "black box", offering limited ability to explicitly identify causal relationships [28]. This lack of transparency can be a challenge when interpretability is crucial. NN models can be more difficult to use in practical applications due to their complexity and require significant computational resources, especially for large datasets. Overfitting is another concern, as NNs can easily fit noise in the training data, leading to poor generalisation [42]. Finally, the development of NN models is largely empirical, with many methodological issues, such as

choosing the right architecture and avoiding overfitting, still unresolved [42].

2.1.1 Deep learning (DL)

DL architectures are characterised by their deep neural networks (DNNs), which consist of numerous layers of interconnected neurons [25]. These networks vary in depth, ranging from tens to hundreds of layers, enabling them to discern intricate patterns and representations from complex datasets. The depth of these networks facilitates the capture of hierarchical features corresponding to different levels of abstraction within the data.

A key distinction between DNNs and traditional NNs is the presence of multiple hidden layers, as illustrated in Figure 2.1. Traditional NNs typically consist of a few hidden layers, whereas DNNs incorporate many hidden layers [30, 31]. This multi-layered architecture allows DNNs to learn increasingly complex and abstract features directly from raw data, enhancing their capability to make accurate predictions or classifications.

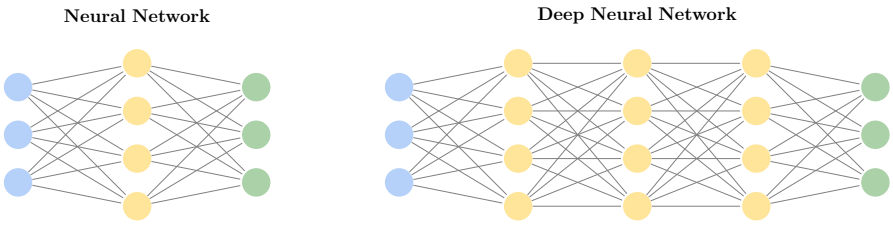


FIGURE 2.1: Difference between NN and DNN architectures

DL systems generally fall into two primary categories based on their architectures: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are designed to process data that is structured as multiple arrays. For example, coloured images are typically represented by three 2D arrays, each corresponding to a different colour channel (red, green and blue) [25]. CNNs are built on key concepts including local connections, shared weights, pooling and multiple layers. The architecture typically includes convolutional layers that detect local features using shared filters across different parts of the input array and pooling layers that merge similar features to create invariance to small shifts and distortions. This hierarchical approach enables CNNs to capture complex patterns and compositional hierarchies, such as edges forming motifs and motifs combining into objects. Inspired by visual neuroscience, CNNs mimic the processing pathways of the human visual system and have been successfully applied to tasks such as image and speech recognition, optical character recognition and object detection.

RNNs, on the other hand, are specialised DL architectures designed for sequential data processing, allowing information from previous steps to influence current predictions [32]. Unlike traditional NNs, which treat inputs and outputs independently, RNNs maintain internal memory to handle sequences effectively, with outputs from previous time steps fed back into the network [38]. This feedback allows RNNs to maintain and update a state vector that captures information about past inputs, which is crucial for understanding temporal dynamics and dependencies in sequential data [25]. However, RNNs face challenges such as the vanishing gradient problem, which limits their ability to learn from long-term dependencies [40]. Specifically, gradients can either grow excessively or diminish as they are back-propagated through each time step, often leading to problems where they explode or vanish over many time steps [25]. To address this, gated recurrent units and long short term memory networks integrate specialised gating mechanisms, which are described in detail in Section 2.1.2.

2.1.2 Recurrent neural networks (RNNs), long short term memory (LSTM) and gated recurrent units (GRUs)

Advancements like LSTM and GRU networks have significantly enhanced RNNs' ability to remember long-term dependencies and handle complex tasks by addressing the vanishing gradient problem [25]. LSTM networks manage long-term dependencies by incorporating mechanisms that regulate the flow of information through the network, allowing them to retain important data over long sequences. GRUs simplify this approach with a more streamlined architecture that effectively manages state updates and reduces gradient issues.

In GRUs, the reset gate r_t controls how much of the previous hidden state $h_{(t-1)}$ should be reset or forgotten when processing the current input x_t, while the update gate z_t determines how much of the new candidate activation $\hat{h}_t$ should be added to the current hidden state $h_{(t-1)}$ [13]. The final hidden state h_t at time-step t is then computed, balancing between retaining previous information and integrating new inputs (Equations 2.1, 2.2, 2.3 and 2.4).

$$r_t = \sigma(W_r[h_{(t-1)}, x_t]) \tag{2.1}$$
$$z_t = \sigma(W_z[h_{(t-1)}, x_t]) \tag{2.2}$$
$$\tilde{h}_t = \tanh(W_h[r_t * h_{(t-1)}, x_t]) \tag{2.3}$$
$$h_t = (1 - z_t) * h_{(t-1)} + z_t * \tilde{h}_t \tag{2.4}$$

In contrast, LSTM networks employ three primary gates — input gate i_t, forget gate f_t and output gate o_t — to regulate information flow through the memory cell C_t [23]. The input gate i_t controls which information from the current input x_t and previous hidden state $h_{(t-1)}$ should be stored in C_t, while the forget gate f_t determines how much of the previous memory cell content $C_{(t-1)}$ should be retained or discarded. The new candidate cell state $\tilde{C}_t$ is computed based on the current input and previous hidden state and the updated memory cell state C_t integrates these components. The output gate o_t then governs which information from C_t should be propagated to the next time-step or used as the network's output, influencing the final hidden state h_t (Equations 2.5, 2.6, 2.7, 2.8, 2.9 and 2.10).

$$i_t = \sigma(W_i[h_{(t-1)}, x_t]) \tag{2.5}$$
$$f_t = \sigma(W_f[h_{(t-1)}, x_t]) \tag{2.6}$$
$$\tilde{C}_t = \tanh(W_C[h_{(t-1)}, x_t]) \tag{2.7}$$
$$C_t = f_t * C_{(t-1)} + i_t * \tilde{C}_t \tag{2.8}$$
$$o_t = \sigma(W_o[h_{(t-1)}, x_t]) \tag{2.9}$$
$$h_t = o_t * \tanh(C_t) \tag{2.10}$$

These gating mechanisms collectively enhance the ability of GRU and LSTM to manage and process sequential data effectively. Figure 2.2 visually compares the architectural differences between RNN, LSTM and GRU, highlighting their distinct approaches to handling sequential information.

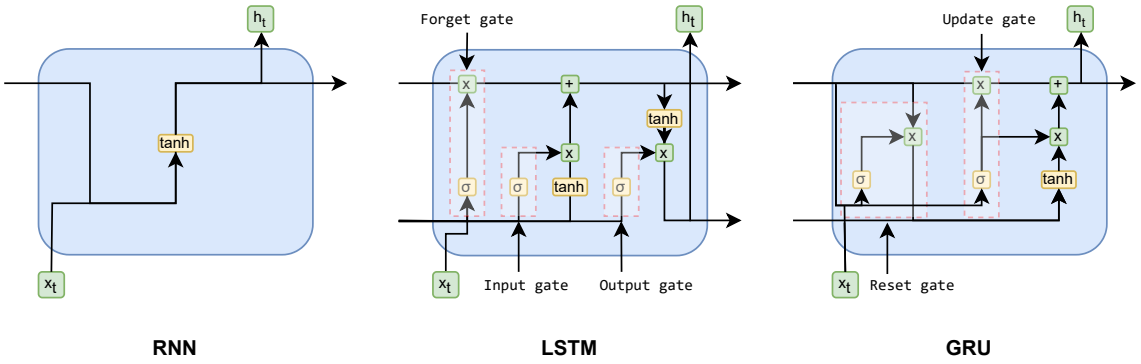


FIGURE 2.2: Architectural comparison of RNN, LSTM and GRU

2.2 Process mining (PM)

PM, a relatively recent field of research, bridges business process modelling with data mining to uncover insights into process variants, bottlenecks and opportunities for enhancement [45]. Central to PM is the use of event logs, which typically consist of case IDs, timestamps and event identifiers, forming sequences where each case contains a series of events identified by event names and timestamps [47]. A case represents a process instance or the subject that undergoes the events/activities. The activities are recorded in the event log, detailing the order in which they occur through timestamps [18]. Additionally, the event log can include attributes at each timestamp, such as resources used and other relevant details.

PM encompasses three primary dimensions: process discovery, conformance checking and process enhancement [44]. Process discovery involves constructing models of actual processes from event logs, which serve as the primary input for PM techniques. Conformance checking ensures alignment between these models and the recorded event logs. Process enhancement focuses on refining models based on insights derived from the process discovery and conformance checking phases.

2.2.1 Predictive process mining (PPM)

PPM leverages event log data generated by information systems during business process executions to predict various business outcomes [8]. Predictive goals can range from predicting the next activity and remaining cycle time to forecasting outcomes. The required features, such as timestamps, resource identifiers, event types and case attributes, are extracted from event logs and used as input for ML algorithms for predictions. These features can be categorised into two types: event-level features and case-level features.

Event-level features are attributes derived from the unique characteristics associated with a specific event within a process. These features are created by extracting event-specific details that offer a comprehensive description of the event itself [47]. Examples of event-level features include the activity and resource labels.

Case-level features, on the other hand, are generated by integrating event-specific and case-specific attributes to deliver a thorough understanding of the corresponding case [47]. Examples of case-level features include the count of occurrence, representing the number of times a particular activity has occurred or a particular role was involved in the process, the time elapsed to the point of prediction and the mean value of numerical features such as the average task duration.

Figure 2.3 illustrates the overall process of PPM, from the initial event log extrac-

tion to the final evaluation of the predictive model. This diagram outlines the key steps involved, including feature extraction, the definition of predictive goals, model building, model training and evaluation.

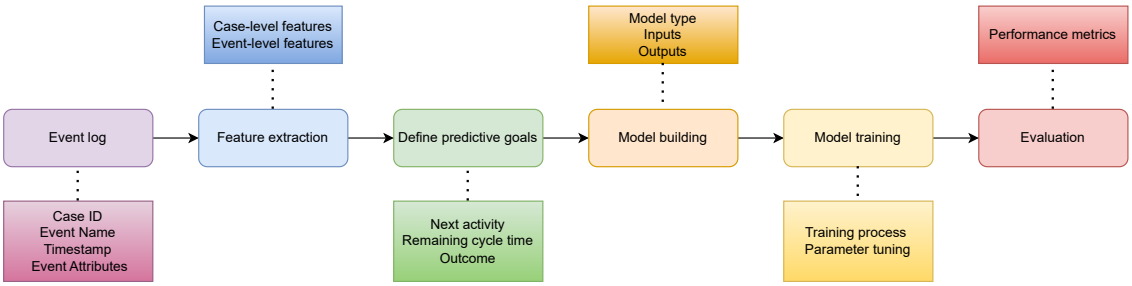


FIGURE 2.3: PPM workflow

2.3 Summary

This chapter establishes a foundational understanding of key concepts necessary for evaluating the comparative effectiveness of PPM and traditional ML methods in predicting ED admissions.

The exploration begins with NNs and their evolution into DL architectures, including RNNs, LSTM networks and GRUs. This foundational knowledge is essential for understanding how DL models, central to both PPM and ML approaches, process and interpret complex datasets.

PM principles are introduced next, highlighting its role in analysing business processes through event logs to uncover inefficiencies and deviations. This sets the stage for understanding PPM, which extends traditional PM by applying predictive analytics to forecast future process states, a critical capability in enhancing healthcare decision-making.

PPM leverages event log data to predict various business outcomes, such as predicting the next activity, remaining cycle time and other outcomes relevant to healthcare settings. By defining predictive goals and extracting relevant features, PPM aims to offer insights that can lead to proactive interventions and improved resource allocation in healthcare environments.

In summary, this chapter provides a comprehensive overview of NNs, DL architectures and PPM techniques, setting a robust foundation for understanding the comparative analysis of PPM and ML methods. This background is crucial for addressing the research question related to enhancing predictive accuracy and operational efficiency in healthcare through advanced data analytics.

Chapter 3

Methodology

The methodology adopted for this research follows the CRoss Industry Standard Process for Data Mining (CRISP-DM) framework [48], illustrated in Figure 3.1. CRISP-DM provides a structured framework comprising six stages: business understanding, data understanding, data preparation, modelling, evaluation and deployment. Each stage is intricately linked to the research objective of mitigating ED overcrowding through the application of ML and PPM techniques:

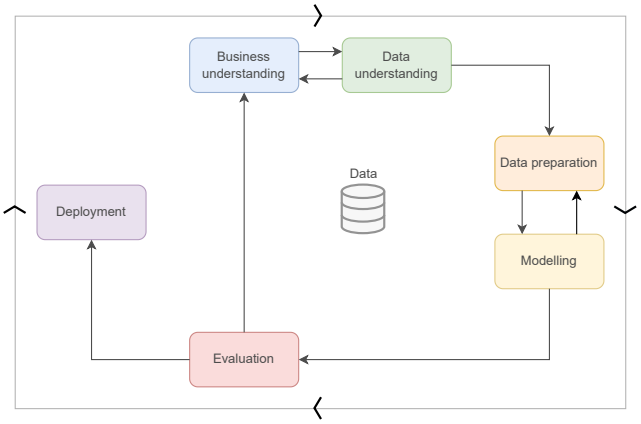


FIGURE 3.1: CRISP-DM framework

1. **Business understanding:** This initial phase involves defining project objectives from a healthcare perspective, particularly focusing on understanding factors contributing to ED overcrowding and how these relate to laboratory test results. This step aligns the research direction with the goal of leveraging ML and PPM for effective prediction and management.

2. **Data understanding:** Here, the emphasis is on exploring the MIMIC-IV dataset, understanding its structure and assessing data quality. This phase ensures that subsequent analyses are based on a comprehensive understanding of the dataset's specifics and healthcare-specific data challenges.

3. **Data preparation:** In this crucial phase, data is selected, cleaned and transformed to facilitate effective modelling. Specific considerations include handling temporal data from sequential medical records to capture dynamic patient conditions related to ED visits.

4. **Modelling:** DL techniques are applied to the prepared dataset, aiming to develop robust predictive models for ED admission status. Methods such as LSTM are explored to capture patterns which are critical in healthcare predictions.

5. **Evaluation:** Model performance is rigorously evaluated against pre-defined metrics to ensure alignment with healthcare objectives, such as accuracy in predicting ED admissions.

6. **Deployment:** In this research, findings are validated with healthcare experts to ensure practical applicability and relevance in clinical settings. Additionally, suggestions are provided on how to integrate the solution into operational workflows.

This structured approach ensures reproducibility throughout each CRISP-DM phase, addressing specific challenges in healthcare data analytics and contributing to effective ED management strategies. The following chapter will delve into these phases in detail, particularly focusing on the data preparation stage across three distinct pipelines: ML, PPM with case-level features and PPM with event-level features. These pipelines will be compared to evaluate their effectiveness in predicting and managing ED overcrowding.

Chapter 4

In-depth analysis of CRISP-DM

This chapter provides a comprehensive analysis of the CRISP-DM methodology, with a specific focus on its application to healthcare data analytics. It details each of the six CRISP-DM phases, illustrating how these phases guide the data mining process and address the unique challenges of healthcare data.

In addition to following the CRISP-DM framework, this research implements three distinct pipelines: ML, PPM with event-level features and PPM with case-level features. These pipelines differ primarily in their approaches to the data preparation stage, tailored to address the complexities of temporal healthcare data and the specific objectives of predicting and managing ED overcrowding.

In particular, this chapter explains how the CRISP-DM methodology is applied to compare the performance of ML and PPM techniques. By examining how each phase of CRISP-DM is utilised, the chapter highlights the steps taken to evaluate and contrast the effectiveness of ML and PPM approaches in classifying ED admissions.

4.1 Business understanding

Understanding the connection between laboratory test results and ED overcrowding is crucial for improving patient care and resource management in healthcare settings. This section delves into the significance of laboratory medical tests and their role in ED overcrowding.

Laboratory medical tests are indispensable tools in modern healthcare, playing a pivotal role in clinical decision-making, patient care and medical research [43]. These tests provide objective data derived from biological samples such as blood, urine and tissue, enabling healthcare providers to confirm diagnoses, monitor disease progression and assess treatment effectiveness based on specific biomarkers or physiological markers present in the samples.

Beyond diagnosis, laboratory tests are critical for continuous disease monitoring and prompt treatment adjustment, particularly in chronic conditions where on-going management relies on precise clinical data [46]. Screening tests facilitate early disease detection, identifying illnesses before symptoms manifest and allowing timely intervention to enhance patient outcomes.

Abnormal or critical values in laboratory tests often signal acute medical conditions or worsening health issues, prompting patient admission to the ED [35]. Immediate monitoring and intervention are essential in such cases to manage patient care effectively. The availability of adequate resources and trained personnel in EDs is crucial to meeting these urgent healthcare needs.

ED overcrowding poses significant challenges, including potential staff shortages and delayed treatment for patients [39]. Predicting ED visits using historical laboratory test data is vital for proactive ED management. Predictive analytics analyse trends and patterns in laboratory test results to forecast patient admissions, facilitating optimal resource allocation, staffing decisions and ED readiness to meet the healthcare demands of emergency patients.

4.2 Data understanding

A thorough understanding of the dataset is crucial for any research project, as it lays the foundation for effective analysis and interpretation. This section provides an overview of the Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset, highlighting its scope, content and significance for research in critical care medicine.

The MIMIC-IV dataset is a comprehensive and freely accessible EHR database designed to support research in critical care medicine and related fields [11, 20, 21]. Developed by researchers at the Massachusetts Institute of Technology (MIT) and Beth Israel Deaconess Medical Center (BIDMC), MIMIC-IV represents a significant advancement in the availability of clinical data for scientific inquiry.

MIMIC-IV contains de-identified health data of around 300 000 patient admissions at BIDMC between 2008 and 2019. It encompasses a wide range of information, including demographics, vital signs, laboratory measurements, medications, procedures, diagnostic codes and clinical notes. The dataset also includes physiological waveforms, such as electrocardiograms (ECGs), arterial blood pressure waveforms and respiratory waveforms, providing rich insights into patients' physiological states.

One of the distinguishing features of MIMIC-IV is its longitudinal nature, allowing researchers to track patients' trajectories over time and investigate complex patterns in their healthcare journeys. Furthermore, MIMIC-IV incorporates data from diverse patient populations, encompassing a variety of medical conditions and treatments, thus facilitating studies across different clinical domains.

4.3 Data preparation

Effective data preparation is a critical step in building reliable predictive models. In this study, the data preparation process is tailored to suit the specific requirements of each pipeline, ensuring that the data is in the optimal format for the chosen algorithms. This section outlines the procedures applied in each pipeline, detailing how the data was processed to enhance the accuracy and robustness of the predictions. The following subsections describe the data preparation steps for each of the three pipelines used in the study: ML, PPM with case-level features and PPM with event-level features.

4.3.1 Pipeline I: ML

Pipeline I refers to the application of traditional ML methods for classifying ED admissions based on a patient's six-month medical test history. Unlike pipelines II and III, which incorporate PPM techniques with case-level and event-level features, respectively, pipeline I focuses solely on conventional data preparation techniques without any PM conventions. The data extraction, data preparation and modeling stages for pipeline I are summarised in Figure 4.1.

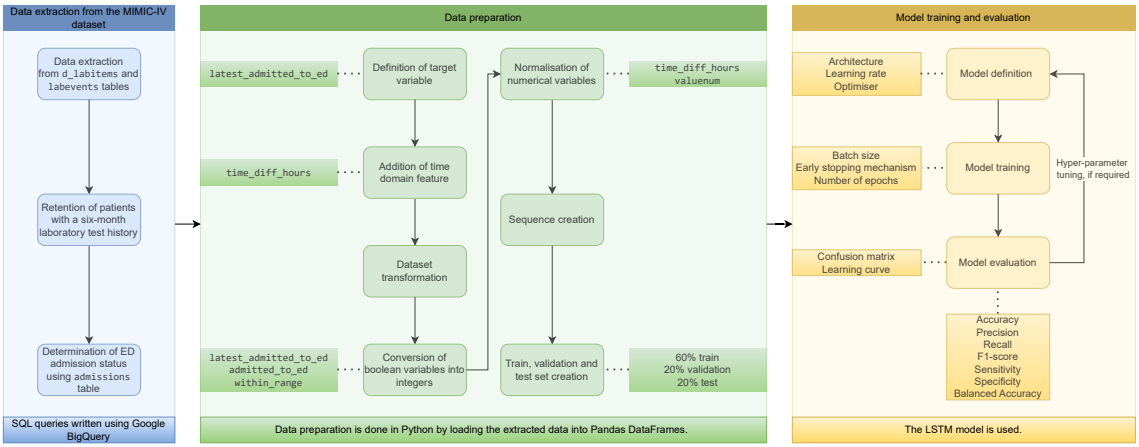


FIGURE 4.1: Data preparation and modelling workflow for pipeline I

Data extraction

Building upon the introduction to the MIMIC-IV dataset in Section 4.2, the analysis focuses on specific data components relevant to the classification of ED admissions using medical laboratory test data. Three tables from the MIMIC-IV dataset were utilised: `labevents`, `d_labitems` and `admissions`, all of which belong to the `hosp` module. The attributes used from each table are summarised in Table 4.1, with detailed descriptions provided in Table 4.2.

TABLE 4.1: Attributes used from MIMIC-IV tables [11, 20, 21]

`labevents`	`d_labitems`	`admissions`
subject_id	itemid	subject_id
hadm_id	label	hadm_id
charttime		edregtime
itemid		edouttime
valuenum		
valueuom		
ref_range_lower		
ref_range_upper		

Data extraction from the MIMIC-IV dataset using Google Cloud BigQuery followed several steps, conducted through SQL queries. Initially, laboratory test information was extracted from the `labevents` and `d_labitems` tables, linking both tables using the `itemid` attribute. This process involved retrieving the following attributes: `subject_id`, `hadm_id`, `charttime`, `valuenum`, `valueuom`, `ref_range_lower`, `ref_range_upper` and `label`. It was ensured that none of the attributes, except for `valueuom`, contained null values. Only the 28 laboratory tests listed in Table A.1 were included, as recommended by a domain expert. These tests were chosen because they are commonly requested by general practitioners in the Netherlands and are most indicative of organ functionality.

The earliest and latest laboratory test times for each patient were then identified using the `subject_id` and `charttime` attributes, ensuring retrieval of patients with a six-month history of laboratory tests.

Finally, using the `admissions` table, it was determined whether each patient was admitted to the ED during each hospital admission, indicated by non-null `edregtime` and

TABLE 4.2: Description of attributes from MIMIC-IV dataset [11, 20, 21]

Attribute	Table	Description
charttime	labevents	It represents the timestamp when the laboratory measurement was recorded, typically corresponding to the time the specimen was collected.
edouttime	admissions	It represents the time that the patient was discharged from the ED.
edregtime	admissions	It represents the time that the patient was admitted to the ED.
hadm_id	admissions, labevents	It is a unique ID signalling a patient's admission to the hospital.
itemid	labevents, d_labitems	It is a unique identifier for a specific laboratory concept.
label	d_labitems	It is a description of the laboratory concept indicated by itemid.
ref_range_lower	labevents	It is the lowest value in the range of normal values for a specific laboratory measurement.
ref_range_upper	labevents	It is the highest value in the range of normal values for a specific laboratory measurement.
subject_id	labevents, admissionss	It is a unique ID identifying a patient.
valuenum	labevents	It represents the numerical value of a laboratory measurement.
valueuom	labevents	It represents the unit of measurement for the laboratory concept.

edouttime values. This information was linked to the laboratory test data using the subject_id and hadm_id attributes.

Target variable definition

After extracting the relevant data using Google Cloud BigQuery, it was loaded into a Pandas DataFrame in Python, where several procedures were conducted. Firstly, the target variable — a boolean indicating whether a patient was admitted to the ED after six months of laboratory tests — was created. This involved adding a corresponding column,

latest_admitted_to_ed, by sorting the DataFrame by subject_id and charttime. For each subject_id, the value of admitted_to_ed was extracted from the row with the most recent charttime to ensure accuracy in representing the latest admission status.

Feature engineering

Next, a time domain feature, time_diff_hours, was added by calculating the time difference in hours between the charttime of the last laboratory test and each prior laboratory test per subject_id. This resulted in a time_diff_hours value of zero for the last laboratory test and positive values for earlier tests. Incorporating this feature is crucial for capturing the temporal context of the laboratory tests, as it helps the model differentiate between recent and older results. Recent tests are often more indicative of a patient's immediate health status, which is critical for predicting ED admissions. By adding time_diff_hours, the model can identify patterns of rapid deterioration or improvement, enhancing its ability to forecast acute events and improve prediction accuracy. This temporal feature enriches the dataset by adding context, allowing the model to detect clinically significant changes that may signal an impending emergency, thereby enhancing overall model performance.

Dataset transformation

For multivariate laboratory test analysis, the DataFrame was transformed to aggregate test results taken at the same time, aligning with the clinical reality where ED admission decisions depend on the combination of laboratory test results. This involved first retrieving unique laboratory test labels into a list. Two columns were created for each laboratory test: one for the numerical result (valuenum) and the other a boolean variable indicating whether the value was within the required range (within_range).

Finally, a pivot table was created by transposing the DataFrame, aggregating all tests performed at the same time into a single row per patient. The label column was removed and the valuenum and within_range values were added to their respective columns. Each row now contained all the laboratory tests performed per charttime and per subject_id. In cases where a test was not performed, the values were filled with -1 to indicate their absence, as this value was not previously used in the DataFrame. This reduced the bias in the data, as other values could significantly affect the distribution of medical laboratory test results. Additionally, the valueuom variable was no longer needed, as all values in each valuenum column now shared the same unit of measurement.

The transformed features and target variable used are summarised in Table 4.3. This table provides an overview of the features incorporated into the model and their descriptions, which are critical for understanding the input and output data used for training and evaluation.

The DataFrame transformation is exemplified in Table 4.4, visually representing the restructuring and organisation of data for further analysis. For better readability, some columns have been omitted. The tables primarily illustrate the transformation concerning the valuenum, valueuom, label and within_range columns, with one row per distinct value of time_diff_hours for each subject_id.

Data preparation for LSTM processing

To prepare the DataFrame for LSTM processing, several steps were undertaken. Firstly, the boolean variables within_range for every laboratory test label, latest_admitted_to_ed and admitted_to_ed were converted to integers. Secondly, the valuenum values for each

TABLE 4.3: Summary of features and target variable used in pipeline I

Feature	**Description**
`time_diff_hours`	This numerical variable represents the time difference in hours between the chart time of the last laboratory test and each preceding test for each `subject_id`, with a value of zero for the last test and positive values for earlier tests.
`valuenum`	This numerical variable represents the value of medical laboratory test results, with one column for each of the 28 laboratory tests.
`within_range`	This boolean variable serves as an indicator of whether the `valuenum` is within the required range, with one column for each of the 28 laboratory tests.
`admitted_to_ed`	This boolean variable indicates whether the patient was admitted to the ED during the hospital admission corresponding to the laboratory tests.
Target variable	**Description**
`latest_admitted_to_ed`	This boolean variable represents whether the patient was admitted to the ED based on the six-month history of medical laboratory tests.

TABLE 4.4: Table transformation process

(A) Table structure before transformation

subject_id	**label**	**time_diff_hours**	**valuenum**	**valueuom**	**within_range**
13015616	Glucose	4163.45	77.0	mg/dL	True
13015616	Potassium	0.0	3.8	mEq/L	True

(B) Table structure after transformation

subject_id	**time_diff_hours**	**Glucose_valuenum**	**Glucose_within_range**	**Potassium_valuenum**	**Potassium_within_range**
13015616	4163.45	77.0	True	-1	-1
13015616	0.0	-1	-1	3.8	True

laboratory test were normalised using Min-Max Scaler based on their respective column values, excluding the -1 placeholder. The minimum and maximum values for each laboratory test across all patients were used for this normalisation. Thirdly, the `time_diff_hours` attribute was also normalised using the Min-Max Scaler. This normalisation ensured that the features had a uniform scale. The data distribution after these transformations was then analysed and summarised in Table 4.5.

To facilitate LSTM processing, sequences were constructed, where the data was converted into tensors and arranged in descending order of `time_diff_hours` per `subject_id`. Pre-padding was applied with a padding value of -2, chosen because it did not appear in the DataFrame. The sequence length was set to the maximum length observed across patients, which was 368. Pre-padding was preferred over post-padding due to its compatibility with LSTM's efficiency [10]. Each patient's sequence included the `valuenum` and `within_range`

TABLE 4.5: Distribution of rows and patients based on the value of the target variable

Value of target variable	Number of patients	Number of rows
True	876	26 026
False	519	16 580

for all laboratory tests conducted, `time_diff_hours` and `admitted_to_ed`, with the target variable `latest_admitted_to_ed` in its own NumPy array. These sequences were then padded to a uniform length of 368, resulting in a final sequence size of (1395, 368, 58), where 1395 represents the number of patients, 368 indicates the sequence length and 58 denotes the number of features included in each sequence.

Model training and evaluation

For training and evaluating the model aimed at classifying ED admissions, the dataset was divided into training (60%), validation (20%) and test sets (20%), with dimensions detailed in Table 4.6. This partitioning strategy ensured that the model was trained on a majority of the data while validating and testing on independent subsets to assess its generalisation capability.

TABLE 4.6: Dimensions of train, validation and test sets

Dataset	Number of samples	Dimensions
Train	837	(837, 368, 58)
Validation	279	(279, 368, 58)
Test	279	(279, 368, 58)

4.3.2 Pipeline II: PPM with case-level features

Pipeline II employs PM techniques combined with traditional ML methods, specifically using case-level features for the classification of ED admissions. In this approach, the six-month medical test history of a patient is analysed at the case level, where each case corresponds to a complete sequence of events (laboratory tests) leading up to an ED admission. This pipeline leverages the aggregated information from these sequences to predict whether a patient will be admitted to the ED. Figure 4.2 provides a summary of the data extraction, preparation and modelling stages for pipeline II.

Data extraction

The data extraction from the MIMIC-IV dataset was performed using Google Cloud BigQuery, following the approach detailed in pipeline I, outlined in Section 4.3.1.

Target variable definition

After extracting the suitable data using Google Cloud BigQuery, it was loaded into a Pandas DataFrame in Python for further processing. Firstly, the target variable — a boolean indicating whether a patient was admitted to the ED after six months of laboratory tests — was created. This involved adding a corresponding column, `latest_admitted_to_ed`, by sorting the DataFrame by `subject_id` and `charttime`. For each `subject_id`, the value

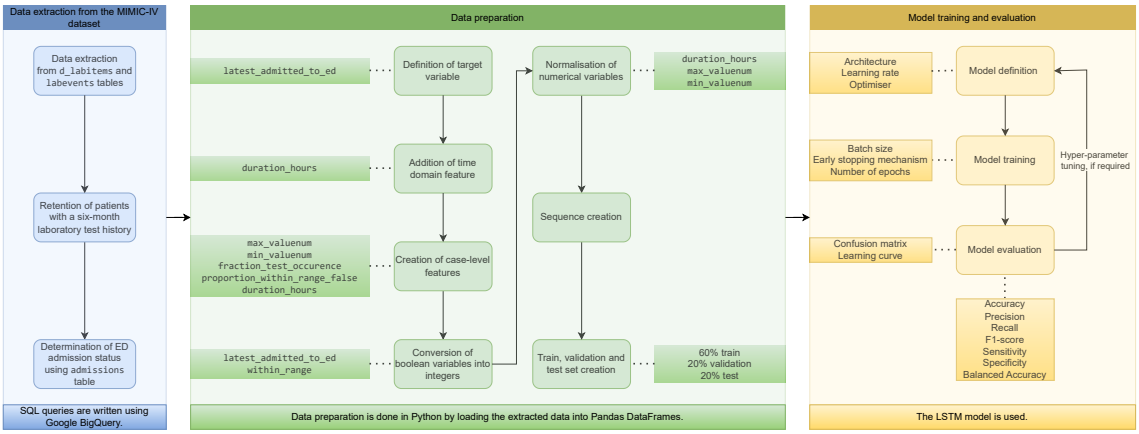


FIGURE 4.2: Data preparation and modelling workflow for pipeline II

of `admitted_to_ed` was extracted from the row with the most recent `charttime` to ensure accuracy in representing the latest admission status.

Feature engineering

Next, a time domain feature, `duration_hours`, was added by calculating the time difference in hours between the `charttime` of the last laboratory test and the first laboratory test per `subject_id`. This feature helps evaluate how the timing and frequency of tests relate to health outcomes, which could enhance the prediction accuracy for ED admissions.

To prepare the data for LSTM processing, several steps were taken under the feature engineering process. 86 case-level features were created for each patient identified by `subject_id` and the target variable was set to `latest_admitted_to_ed`. The features and the target variable are detailed in Table 4.7.

Data preparation for LSTM processing

Since some patients did not undergo all tests, there were a significant number of null values in the minimum and maximum values for each laboratory test. Firstly, -1 was used as a placeholder value for these null items. Secondly, the boolean variables `latest_admitted_to_ed` and `within_range` were converted to integers to ensure compatibility with the LSTM model. Thirdly, the `duration_hours`, `max_valuenum` and `min_valuenum` values were normalised using the Min-Max Scaler, with the -1 placeholder excluded from this normalisation process. This step was crucial to ensure that all variables had a uniform scale. The data distribution following these transformations is shown in Table 4.8, which reflects a 3:2 ratio between classes, indicating that for every two non-ED patients, there are approximately three ED patients.

To facilitate LSTM processing, sequences were constructed by converting the data into tensors. The sequence length was set to one, as there were only aggregated features per `subject_id` and no padding was required. Each sequence contained the 86 case-level features, resulting in a final sequence size of (1395, 1, 86), where 1395 represents the number of patients, 1 indicates the sequence length and 86 denotes the number of features included in each sequence. The target variable `latest_admitted_to_ed` for each `subject_id` was stored in a separate NumPy array.

TABLE 4.7: Summary of features and target variable used in pipeline II

Feature	**Description**
`duration_hours`	This numerical variable represents the time difference in hours between the `charttime` of the first and last laboratory tests for each `subject_id`, indicating the time span over which the tests were conducted.
`max_valuenum`	This numerical variable represents the maximum value of each laboratory test for each `subject_id`. There is one column for each of the 28 laboratory tests.
`min_valuenum`	This numerical variable represents the minimum value of each laboratory test for each `subject_id`. There is one column for each of the 28 laboratory tests.
`fraction_test_occurrence`	This numerical variable represents the proportion of times each specific laboratory test was conducted relative to the total number of tests performed for each `subject_id`. There is one column for each of the 28 laboratory tests.
`proportion_within_range_false`	This boolean variable indicates the proportion of times the `within_range` variable had a false value for each `subject_id`.
Target variable	**Description**
`latest_admitted_to_ed`	This boolean variable represents whether the patient was admitted to the ED based on the six-month history of medical laboratory tests.

TABLE 4.8: Distribution of rows and patients based on the value of the target variable

Value of target variable	**Number of patients**	**Number of rows**
True	876	876
False	519	519

Model training and evaluation

The data was then split for training and evaluation: 60% of the data was allocated to the training set, 20% to the validation set and 20% to the test set, ensuring consistency with the other pipelines. The train, validation and test sets have the dimensions outlined in Table 4.9.

4.3.3 Pipeline III: PPM with event-level features

Pipeline III further extends the PPM approach by focusing on event-level features for classification. Instead of aggregating data at the case level, this pipeline examines the individual events (laboratory tests) within the six-month medical history, capturing the

TABLE 4.9: Dimensions of train, validation and test sets

Dataset	Number of samples	Dimensions
Train	837	(837, 1, 86)
Validation	279	(279, 1, 86)
Test	279	(279, 1, 86)

temporal dynamics and detailed characteristics of each event. By incorporating these granular features into the ML model, pipeline III aims to improve the accuracy of predicting ED admissions. A summary of the data extraction, preparation and modelling phases for pipeline III is depicted in Figure 4.3.

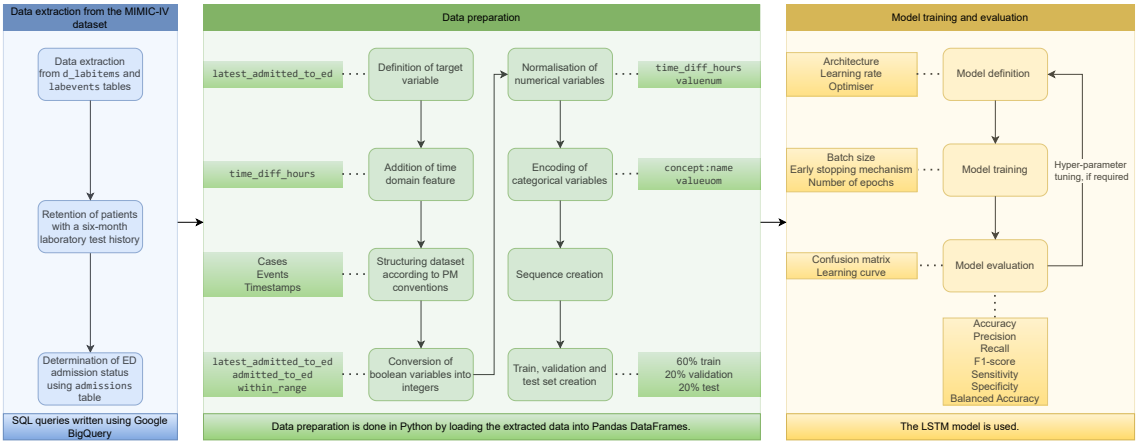


FIGURE 4.3: Data preparation and modelling workflow for pipeline III

Data extraction

Again, the data extraction from the MIMIC-IV dataset was performed using Google Cloud BigQuery, following the approach detailed in pipeline I, outlined in Section 4.3.1.

Target variable definition

After extracting the required data using Google Cloud BigQuery, it was loaded into a Pandas DataFrame in Python and several procedures were conducted. Firstly, the target variable — a boolean indicating whether a patient was admitted to the ED after six months of laboratory tests — was created. This involved adding a corresponding column, `latest_admitted_to_ed`, by sorting the DataFrame by `subject_id` and `charttime`. For each `subject_id`, the value of `admitted_to_ed` was extracted from the row with the most recent `charttime` to ensure accuracy in representing the latest admission status.

Feature engineering

Next, a time domain feature, `time_diff_hours` was added by calculating the time difference in hours between the `charttime` of the last laboratory test and each prior laboratory test per `subject_id`. This resulted in a `time_diff_hours` value of zero for the last laboratory test and positive values for earlier tests. Again, incorporating the `time_diff_hours` feature is vital for capturing the temporal context of laboratory tests. It helps the model distinguish between recent and older results, which is crucial for assessing immediate health

status and predicting ED admissions. This feature allows the model to detect patterns of rapid change, improving its ability to forecast acute events and overall prediction accuracy.

Structuring data according to PM conventions

The columns were then labelled to follow PM conventions as follows: `subject_id` as `case:concept:name`, `charttime` as `time:timestamp` and `label` as `concept:name`. In this context, `case:concept:name` represents the unique case identifier, `time:timestamp` represents the time at which the event took place and `concept:name` represents the event that occurs. The dataset then contains the columns: `case:concept:name`, `time:timestamp`, `concept:name`, `time_diff_hours`, `valuenum`, `valueuom`, `within_range`, `admitted_to_ed` and `latest_admitted_to_ed`. Here, `time_diff_hours`, `valuenum`, `valueuom`, `within_range` and `admitted_to_ed` are the additional attributes and`latest_admitted_to_ed` is the target variable. The features and the target variable are described in detail in Table 4.10.

TABLE 4.10: Summary of features and target variable used in pipeline III

Feature	**Description**
`concept:name`	This categorical variable represents the medical laboratory test taken, indicating the event in an event log.
`time_diff_hours`	This numerical variable represents the time difference in hours between the chart time of the last laboratory test and each preceding test for each `subject_id`, with a value of zero for the last test and positive values for earlier tests
`valuenum`	This numerical variable represents the value of medical laboratory test result.
`valueuom`	This categorical variable represents the unit of measurement for the medical laboratory test.
`within_range`	This boolean variable serves as an indicator of whether the `valuenum` is within the required range.
`admitted_to_ed`	This boolean variable indicates whether the patient was admitted to the ED during the hospital admission corresponding to the laboratory tests.
Target variable	**Description**
`latest_admitted_to_ed`	This boolean variable represents whether the patient was admitted to the ED based on the six-month history of medical laboratory tests.

Data preparation for LSTM processing

To prepare the DataFrame for LSTM processing, several steps were undertaken. Firstly, the boolean variables `within_range`, `latest_admitted_to_ed` and `admitted_to_ed` were

converted to integers. Secondly, the `valuenum` and `time_diff_hours` values were scaled using the Min-Max Scaler to ensure they were within the range between zero and one. Thirdly, the `concept:name` and `valueuom` were encoded using the Label Encoder to ensure proper processing by LSTM. The data distribution was then analysed and summarised in Table 4.11.

TABLE 4.11: Distribution of rows and patients based on the value of the target variable

Value of target variable	Number of patients	Number of rows
True	876	227 572
False	519	146 822

To facilitate LSTM processing, sequences were constructed, where the data was transformed into tensors and arranged in descending order of `time_diff_hours` per `subject_id`. Pre-padding was applied with a padding value of -1, chosen because it did not appear in the DataFrame. The sequence length was set to the maximum length observed across patients, which was 3253. Pre-padding was preferred over post-padding due to its compatibility with LSTM's efficiency [10]. These sequences were then padded to a uniform length of 3253, resulting in a final sequence size of (1395, 3253, 6), where 1395 represents the number of patients, 3253 indicates the sequence length and 6 denotes the number of features included in each sequence. The target variable `latest_admitted_to_ed` for each `subject_id` was stored in a separate NumPy array.

Model training and evaluation

For training and evaluating the model aimed at classifying ED admissions, the dataset was divided into training (60%), validation (20%) and test sets (20%), with dimensions detailed in Table 4.12. This partitioning strategy ensured that the model was trained on a majority of the data while validating and testing on independent subsets to assess its generalisation capability. The strategy was carefully designed to match the approach used in the other pipelines, maintaining consistency across different methodologies and ensuring that comparisons of model performance are valid and reliable.

TABLE 4.12: Dimensions of train, validation and test sets

Dataset	Number of samples	Dimensions
Train	837	(837, 3253, 6)
Validation	279	(279, 3253, 6)
Test	279	(279, 3253, 6)

4.4 Modelling

To ensure a fair comparison, the same model architecture was applied across all three pipelines. The design was carefully crafted to optimise performance for the classification task. The model began with a masking layer to ignore padded values in the sequences (applied only to pipelines I and III, where the sequence length was greater than one). This was followed by an LSTM layer with 16 hidden units, tailored to efficiently handle the sequential nature of the input data. The choice of LSTM over GRU was made based on

its proven effectiveness in retaining long-term memory, as LSTM networks generally perform better with complex sequential data and long-term dependencies [17]. Additionally, increasing the number of hidden units or LSTM layers introduced fluctuations in the losses and widened the gap between validation and training losses, suggesting overfitting. This aligns with findings from previous studies, which have shown that adding more hidden units or layers can lead to overfitting, as the model begins to memorise the training data at the expense of generalising to new data [3, 16]. The final layer was a dense layer with a sigmoid activation function, appropriate for binary classification tasks, where the output represents the probability of an ED admission.

Hyper-parameter tuning involved extensive experimentation with the model's architecture, learning rate, optimiser, loss function, number of epochs, batch size and early stopping patience. The details of the hyper-parameters tuned are summarised in Table 4.13. The ADAM optimiser with a fine-tuned learning rate of 0.0001 and binary cross-entropy loss function were selected as optimal choices for the task.

Training proceeded over 350 epochs with a batch size of 16, leveraging early stopping with a patience of 10 epochs to monitor validation loss and prevent overfitting. Throughout training, the model's performance was continuously evaluated using the selected metrics (Table 4.14), with close monitoring of training and validation losses to ensure effective learning progress and model robustness.

4.5 Evaluation

The evaluation metrics selected to measure the performance of the models are detailed in Table 4.14. These metrics were chosen specifically to assess the models' ability to predict ED admissions based on the input features. True positives, false positives, true negatives and false negatives are specifically shown in confusion matrices, providing a detailed breakdown of the model's classification performance. For precision, recall and F1-score, the weighted average is used to account for class imbalance.

In addition to these metrics, learning curves are employed to analyse the model's performance across different stages of training, providing insights into potential overfitting or underfitting issues. The confusion matrices offer a comprehensive view of the model's classification accuracy and are key to understanding how well the models distinguish between the different classes.

The primary goal is to determine whether PPM outperforms standalone ML models in predicting ED admissions. The detailed evaluation results, including the learning curves and confusion matrices, are presented in Chapter 5.

4.6 Deployment

The deployment phase focuses on validating the model's findings with healthcare experts and developing a practical implementation strategy for real-world application. This begins with adapting data from the healthcare system — such as test timestamps, test results, patient admission statuses and compliance with acceptable ranges — into a format that optimises the model's performance based on pre-defined metrics. This ensures the model integrates effectively into clinical settings and that the data is actionable for both real-time and batch processing.

Subsequently, the model will be integrated into existing healthcare systems using appropriate tools and technologies, potentially as a web service or within healthcare applications. This integration includes ensuring that the model's outputs are useful and actionable

Table 4.13: Parameters tuned

Parameter	Value	Description
`rnn_layer_type`	LSTM	It specifies the type of RNN layer used.
`units`	16	It specifies the number of hidden units in the `rnn_layer_type`.
`learning_rate`	0.0001	It specifies the size of the steps taken during the optimisation process to minimise the loss function. It controls how much to change the model in response to the estimated error each time the model weights are updated.
`num_epochs`	350	It specifies the number of complete passes through the entire training dataset. Each epoch means that the model has seen every training sample once.
`batch_size`	16	It specifies the number of training samples used to calculate each update to the model's parameters.
`patience`	10	This is a hyper-parameter used with early stopping, which defines the number of epochs with no improvement in the monitored metric, in this case the validation loss, after which training will be stopped.

within clinical workflows. Practical recommendations for implementation involve developing user-friendly interfaces, integrating with EHR systems and establishing protocols for utilising model predictions in decision-making. A training program will also be provided to ensure healthcare staff can effectively use the system and interpret its outputs.

Ongoing performance monitoring is crucial, with regular assessments of the model's accuracy and impact. Feedback from stakeholders will guide necessary adjustments to enhance relevance and effectiveness, while continuous maintenance will involve updating the model with new data and adapting to evolving healthcare practices to ensure long-term value.

TABLE 4.14: Evaluation metrics for model performance [14]

Metric	**Description**
True positives	It represents the number of correctly predicted positive instances.
False positives	It represents the number of incorrectly predicted positive instances.
True negatives	It represents the number of correctly predicted negative instances.
False negatives	It represents the number of incorrectly predicted negative instances.
Accuracy	It is the proportion of correctly classified instances.
Precision	It is the proportion of true positive predictions among all positive predictions. Weighted average precision considers precision values for each class, weighted by the number of true instances for each class.
Recall	It is the proportion of true positive predictions among all actual positive instances. Weighted average recall takes into account recall values for each class, weighted by the number of true instances for each class.
F1-score	It is the harmonic mean of precision and recall. Weighted average F1-score computes F1-score values for each class, weighted by the number of true instances for each class.
Sensitivity	It is the proportion of true positive predictions among all actual positive instances.
Specificity	It is the proportion of true negative predictions among all actual negative instances.
Balanced accuracy	Balanced accuracy is the average of sensitivity and specificity.

Chapter 5

Results

This chapter presents the evaluation results of the predictive models for forecasting ED admissions based on medical laboratory test data. It compares the performance of standalone ML and PPM models, highlighting key insights from their data requirements, learning curves, evaluation metrics and confusion matrices.

Section 5.1 addresses the first sub-research question, discussing the specific data requirements and format differences between the three pipelines. Section 5.2 focuses on the second sub-research question, examining the learning abilities of the DL models. Section 5.3 and Section 5.4 explore the predictive performance differences in terms of pre-defined metrics to answer the third sub-research question. Finally, Section 5.5 answers the main research question.

5.1 Data requirements and format differences

The models in this study process sequences of laboratory tests taken by each patient, yet they differ significantly in how these tests are represented and utilised:

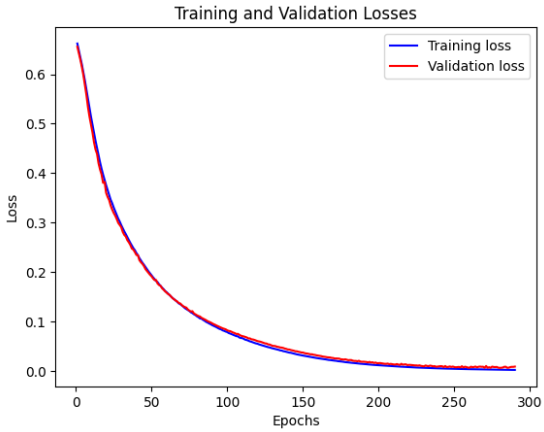
1. **ML model:** Each row in the DataFrame aggregates all laboratory tests taken at a single timestamp, enabling the model to train on the combination of laboratory tests at each point in time.
2. **PPM model with case-level features:** This model aggregates features at the case level but does not capture temporal changes in laboratory test results.
3. **PPM model with event-level features:** Here, each row in the DataFrame represents a single laboratory test without considering other tests taken simultaneously.

The ML model mirrors clinical practice, where patient admissions often rely on the collective insight provided by multiple tests rather than individual values. However, in the PPM model with case-level features, valuable information about trends and fluctuations over time is excluded, which is critical for making precise clinical decisions based on evolving patient conditions. Similarly, the PPM model with event-level features overlooks the combined effect of multiple tests at the same timestamp, which can be crucial for accurately predicting ED admissions based on comprehensive patient profiles.

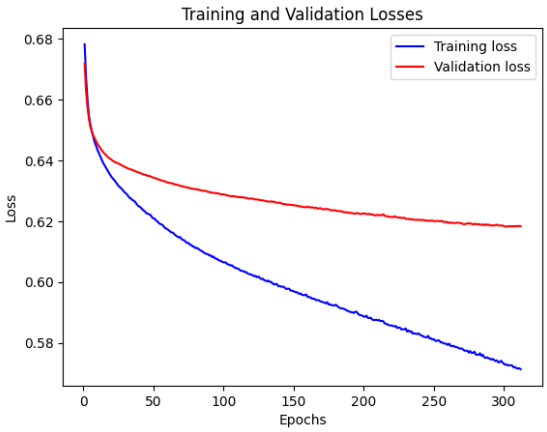
5.2 Learning curves

Figure 5.1 presents the learning curves for the standalone ML model, the PPM model with case-level features and the PPM model with event-level features. The ML model and

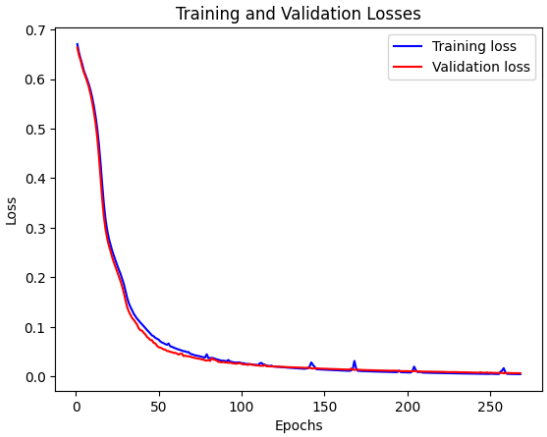
the PPM model with event-level features effectively integrate comprehensive datasets and account for temporal considerations, capturing the complexities of the data. In contrast, the PPM model with case-level features, which aggregates data at the case level, operates on a reduced dataset size. Despite the smaller dataset, this model provides valuable insights into the impact of case-level aggregation on learning dynamics. The learning curves for all three models highlight their respective performances and demonstrate how each approach navigates the temporal and data complexities involved in predicting ED admissions.



(A) ML



(B) PPM: case-level



(C) PPM: event-level

FIGURE 5.1: Comparison of learning curves for pipelines I, II and III

- The standalone ML model (Figure 5.1a) begins with both training and validation losses around 0.70. Both losses decrease steadily over 300 epochs, exhibiting a smooth and consistent downward trend. The training and validation losses closely follow each other throughout the training period, with minimal fluctuations and both losses stabilise near zero. This close alignment between the training and validation losses suggests effective training without significant overfitting or underfitting.

- The PPM model with case-level features (Figure 5.1b) starts with both training and validation losses around 0.68. The training loss steadily decreases over approximately 300 epochs, following a smooth and consistent downward trend before stabilising at a lower value. In contrast, the validation loss also decreases but at a slower

rate, eventually flattening out. As the epochs progress, the widening gap between the training and validation losses suggests potential overfitting, where the model is learning the training data too well but not generalising as effectively to the validation data. Despite this, the continuous decrease in validation loss indicates that the model is still learning, albeit at a slower pace than the training set.

- The PPM model with event-level features (Figure 5.1c) starts similarly with training and validation losses around 0.70 and shows a rapid decrease in the first 50 epochs. The losses continue to decrease gradually and stabilise near zero at around 250 epochs. There are minor fluctuations in the training loss curve, particularly after 50 epochs. Despite these fluctuations, the training and validation losses exhibit good convergence, indicating effective training. However, the presence of minor fluctuations suggests slightly less stability compared to the standalone ML model.

While all models demonstrate effective learning capabilities, they reveal distinct differences in stability and generalisation. The standalone ML model exhibits smooth and consistent training dynamics with minimal fluctuations and a close alignment between training and validation losses, indicating excellent generalisation and minimal overfitting. In contrast, the event-level PPM model, while converging effectively, shows minor fluctuations in training loss, suggesting slightly less stability, potentially due to the complexities of handling temporal features. The PPM model with case-level features, on the other hand, shows a steady decrease in training loss but a widening gap between training and validation losses. This indicates potential overfitting, likely due to its reduced dataset size, which impacts its ability to generalise well to new, unseen data. This raises concerns about its reliability in real-world applications, especially when data is limited.

5.3 Evaluation metrics

Table 5.1 summarises key evaluation metrics including precision, recall and F1-score, computed using weighted averages to account for class imbalances. These metrics provide insights into each model's performance in accurately classifying patient admissions into ED and non-ED categories.

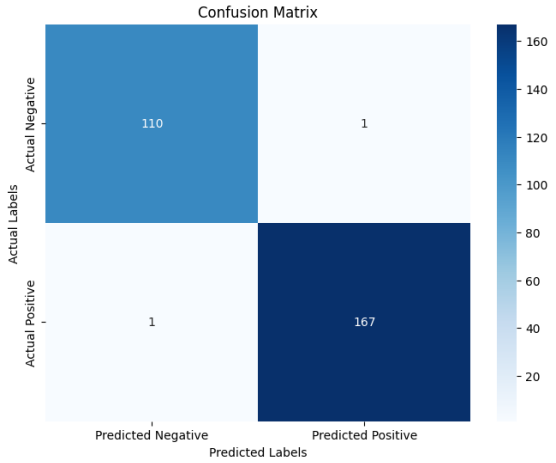
TABLE 5.1: Comparison of performance metrics for pipelines I, II and III

Metric	**ML**	**PPM: case-level**	**PPM: event-level**
Accuracy	99.3%	60.6%	99.6%
Precision	99.3%	58.0%	99.6%
Recall	99.3%	60.6%	99.6%
F1-Score	99.3%	56.3%	99.6%
Sensitivity	99.4%	85.1%	99.4%
Specificity	99.1%	23.4%	100.0%
Balanced accuracy	99.3%	54.3%	99.7%

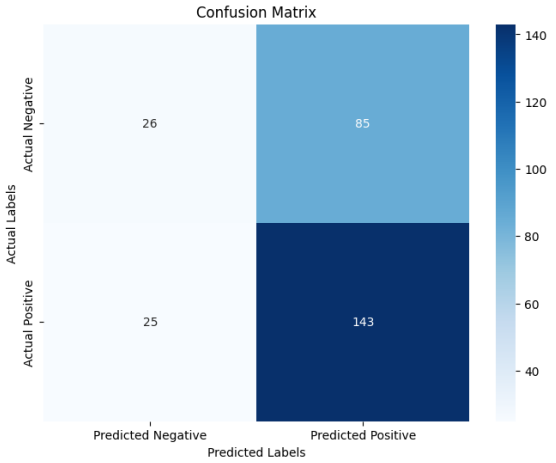
The table highlights that the PPM model with event-level features outperforms the other models across most metrics, achieving near-perfect scores. The standalone ML model shows comparable performance, with only a 0.3% to 0.9% lower value in most metrics. In contrast, the PPM model with case-level features demonstrates significantly lower performance, suggesting that there is substantial room for improvement in handling imbalanced data and capturing complex patterns.

5.4 Confusion matrices

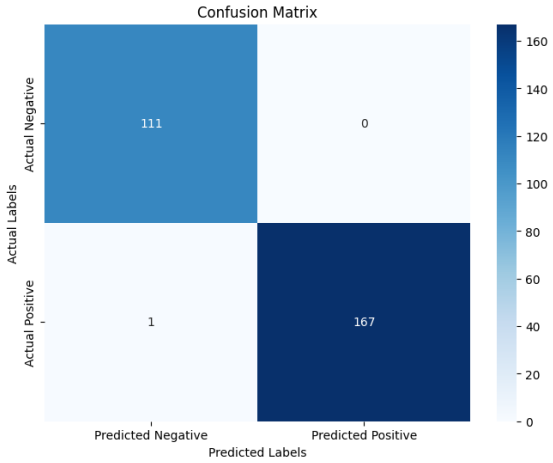
To further explain model performance, Figure 5.2 presents the confusion matrices, providing a detailed breakdown of the predictions made by the models compared to the actual ground truth labels across the two classes.



(A) ML



(B) PPM: case-level



(C) PPM: event-level

FIGURE 5.2: Comparison of confusion matrices for pipelines I, II and III

The confusion matrices reveal the following:

- **Sensitivity:** The models exhibit high sensitivity or recall rates of 99.4%, 85.1% and 99.4% for the ML, PPM case-level and PPM event-level models, respectively.
- **False negatives and false positives:**
  - The PPM model with case-level features captures 85.1% of positive instances accurately, with 14.9% classified as false negatives. In contrast, the PPM model with event-level features demonstrates a sensitivity of 99.4%, correctly identifying 99.4% of positive instances and missing only 0.6% (false negatives). Similarly, the standalone ML model achieves a recall rate of 99.4%, with 0.6% of positive instances being incorrectly classified as negative.

– The ML model has one false negative, indicating that one positive instance is incorrectly classified as negative; the same applies for the PPM model with event-level features. The PPM model with case-level features shows 25 false negatives out of 168 positive instances, along with 85 false positives out of 111 negative instances. In contrast, the PPM model with event-level features exhibits a low false positive rate, with none of the negative instances incorrectly predicted as positive.

5.5 Classifying ED admissions with ML and PPM

The results highlight the effectiveness of both the standalone ML model and the PPM model with event-level features in identifying cases requiring urgent attention or admission to the ED based on medical laboratory tests. With sensitivities of 99.4% and specificities of 99.1% and 100% respectively, these models ensure that nearly all true positive cases are identified while maintaining nearly perfect accuracy in identifying true negative cases. However, the PPM model with case-level features exhibits a high false positive rate of 76.6%, suggesting a potential risk of ED overcrowding due to over-prediction of positive cases. This underscores the importance of balancing sensitivity and specificity to optimise the overall utility of predictive models in clinical settings.

The comparative analysis reveals that the PPM model with event-level features offers comparable performance to the standalone ML model in predicting ED overcrowding using medical laboratory test data. Specifically, the PPM model with event-level features does not significantly improve performance over the standalone ML model, as indicated by the minimal difference in false positives. Both models demonstrate high sensitivity and specificity, highlighting their effectiveness in predicting ED admissions. However, the standalone ML model significantly outperforms the PPM model with case-level features in terms of accuracy, precision and recall, indicating that while PPM with event-level features can achieve performance comparable with traditional ML methods, PPM with case-level features does not meet the same level of effectiveness.

In summary, while the standalone ML model and PPM model with event-level features both demonstrate strong predictive capabilities, the choice of model should consider the specific clinical context and data availability. The higher susceptibility to overfitting in the PPM model with case-level features highlights the challenges of working with smaller datasets and the need for more robust methods to manage class imbalance.

Chapter 6

Discussion

Existing literature predominantly integrated laboratory tests with other clinical factors rather than relying solely on laboratory tests or extensive patient histories, underscoring the complexity of predictive modelling in clinical settings. For example, [19] utilised the first routinely collected tests during hospital stays, while [22] focused on the latest laboratory tests within the preceding 72 hours, combined with vital signs, to predict unplanned ICU transfers. [15] incorporated clinical and venous biochemical measurements over multiple periods to predict hospital mortality, indicating the use of more than just laboratory tests. [29] identified the first blood test around ED admission to predict in-hospital mortality, incorporating personal information like age and National Early Warning Scores. [33] utilised initial haematology and biochemistry tests, patient demographics and hospital outcomes to predict in-hospital mortality. These studies collectively highlight the challenges and importance of considering temporal changes in clinical data for accurate predictive modelling, as solely extensive laboratory test medical histories were not used.

Figure 6.1 illustrates the dynamic changes in laboratory test values over time for a patient, highlighting the significance of these temporal changes in predictive modelling. It demonstrates that different combinations of laboratory tests are conducted at each timestamp, underscoring the variability in data availability over time.

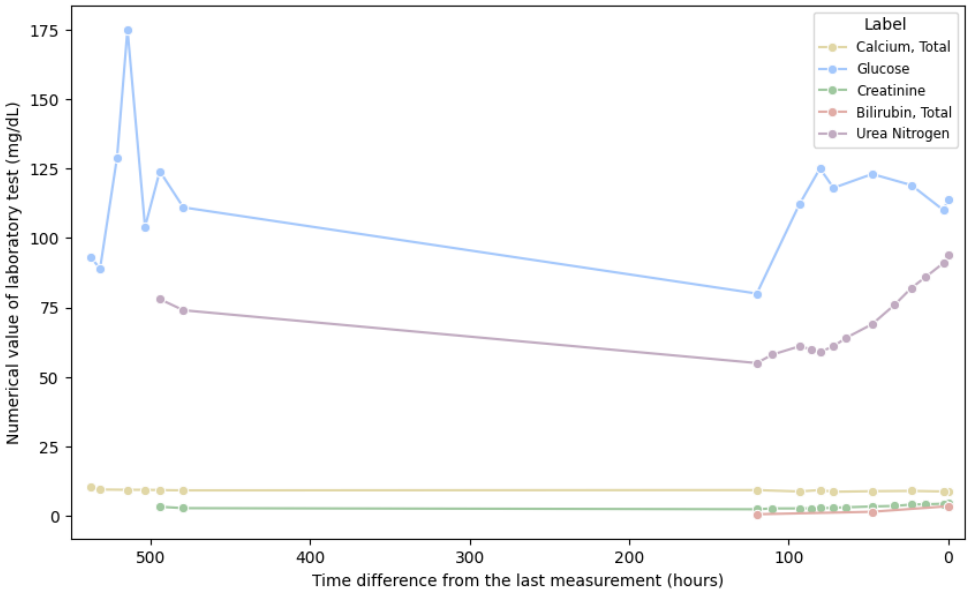


FIGURE 6.1: Illustration of temporal changes in laboratory tests for a patient

The standalone ML model and the PPM model with event-level features exhibit similar performance, primarily because both approaches leverage the same underlying data, but in different formats. In the standalone ML model, each row aggregates all laboratory tests taken at a single timestamp, providing a combined view of multiple test results which reflects real clinical practice. Similarly, the event-level PPM model represents each laboratory test result individually but maintains the temporal component of the data, which contributes to its comparable performance.

Notably, the PPM model with event-level features had just one less false positive than the standalone ML model. In the context of medical decision-making, this difference is not particularly significant. False positives in this scenario refer to instances where the model predicts that a patient should be admitted to the ED when it might not be strictly necessary. However, considering the critical nature of healthcare, it is usually wiser to take a cautious approach. Admitting a patient to the ED when it might not be necessary (a false positive) is less risky than failing to admit a patient who truly needs urgent care (a false negative). False negatives can lead to serious consequences, such as the worsening of a patient's condition due to lack of timely intervention. Therefore, the slightly higher number of false positives in the standalone ML model is acceptable and even favourable, as it prioritises patient safety by reducing the risk of missing critical ED admissions. This implies that the PPM model with event-level features does not necessarily outperform the standalone ML model, as the difference in false positives does not significantly affect the overall effectiveness in the context of patient safety and care.

In contrast, the PPM model with case-level features aggregates all test results per patient, which fails to capture the temporal variations and fluctuations of individual tests over time. This aggregation limits the model's ability to reflect the dynamic nature of real-world data, resulting in a noticeable performance drop compared to the ML and event-level PPM approaches. The similar performance of the standalone ML model and the event-level PPM model underscores the importance of preserving temporal details in the data, while the case-level approach's aggregation of data highlights its limitation in capturing essential temporal changes, leading to less effective predictive modelling.

Additionally, performing multiple tests on a specimen of blood taken at one point in time can be advantageous. Detecting an abnormality in one test result allows other tests to either confirm or exclude the presence of certain diseases, potentially reducing the necessity for follow-up investigations [6]. This underscores the importance of considering combinations of laboratory tests, highlighting the effectiveness of the standalone ML model in leveraging such combined information. By accurately diagnosing conditions through comprehensive testing, the need for repeat visits and prolonged ED stays can be minimised, thereby contributing to alleviating ED overcrowding.

Furthermore, a comprehensive six-month patient history proves crucial for advanced prediction capabilities, enabling proactive preparation of ED resources and efficient patient volume management. The primary objective remains optimising ED utilisation by minimising both false positives and false negatives. Patients with multiple hospital admissions offer a richer historical context, aiding in more accurate predictions regarding the necessity of ED care. However, the dataset primarily includes medical data collected during hospital admissions, excluding information from general practitioner visits and pre-hospitalisation tests. This exclusion limits the model's ability to capture a comprehensive patient medical history, potentially missing crucial health indicators and trends that could affect predictive accuracy.

Additionally, the dataset exhibits inherent class imbalance, with 60% of patients admitted to the ED and the remaining 40% not admitted. This imbalance poses challenges

in effectively training the models, impacting their ability to generalise well across both ED and non-ED cases and influencing performance metrics such as sensitivity and specificity. Specifically, the variation in the number of records between ED and non-ED cases (see Table 4.5, Table 4.8 and Table 4.11) complicates model training. ED patients typically have more tests conducted than non-ED patients, leading to a higher volume of data for those admitted to the ED. This discrepancy means that models trained on such data may become biased towards patterns specific to the ED patients due to the sheer volume of test results they generate.

For instance, ED patients often undergo a broader range of diagnostic tests and more frequent testing compared to non-ED patients, as the nature of emergency care requires rapid and thorough assessment to address acute or uncertain clinical situations [12]. This results in a larger and potentially more complex dataset for the ED group, including various laboratory test results, imaging studies and clinical observations. This increased volume of data can introduce variability in model performance, as models may become overfitted to the more extensive dataset of ED patients or underperform on the smaller dataset of non-ED patients. Balancing techniques were considered to address this imbalance; however, these approaches either risk significant information loss or introduce bias into the data.

This issue is especially pronounced for the PPM model with case-level features. The case-level approach aggregates all test results per patient, leading to a dataset with significantly fewer rows compared to the standalone ML and event-level PPM approaches. This reduction in data volume contributes to a bias in the model's predictions, with a tendency to classify most instances as positive due to the disproportionate representation of data between the two classes. Furthermore, LSTM models are prone to overfitting in small datasets [40]. Consequently, the model is more inclined to classify patients as being admitted to the ED even when it may not be necessary, resulting in an increased number of false positives. In contrast, the standalone ML model and the PPM model with event-level features manage the data in ways that better address class imbalance and maintain performance, even with a maintained 3:2 ratio between ED and non-ED patients.

In conclusion, the standalone ML model and the PPM model with event-level features demonstrate comparable performance and better stability in predicting ED admissions, owing to their effective handling of temporal data and mitigation of class imbalance issues. However, the PPM model with case-level features shows reduced predictive accuracy due to its inability to capture temporal changes, highlighting its limitations in accurately reflecting real-world clinical scenarios and leading to less effective predictive modelling in the ED context.

Chapter 7

Conclusions and future work

The study aims to evaluate the predictive capabilities of both standalone ML and PPM models for classifying ED admissions using a comprehensive six-month history of laboratory tests. By focusing on sequential medical data, the goal is to determine whether PPM models can outperform traditional ML models in classifying ED admissions and thereby help mitigate ED overcrowding. Effective prediction would enable medical practitioners to anticipate admissions better and prepare the ED more efficiently, optimally allocating resources in advance. The research uses the MIMIC-IV dataset, concentrating on laboratory tests conducted during hospital admissions and analysing dynamic changes over time as well as combinations of tests at specific timestamps.

The study begins by examining the distinct data requirements and format differences between PPM techniques and standalone ML methods in Section 7.1, highlighting how these factors impact the applicability and performance of each approach. Section 7.2 focuses on comparing the learning curves of the two methods, evaluating their training dynamics and convergence behavior. The predictive performance of the models is then assessed in Section 7.3, where the accuracy and generalisation capabilities of PPM techniques are compared with those of standalone ML models. Finally, Section 7.4 addresses the main research question by evaluating the overall effectiveness of PPM models relative to traditional ML models, discussing their relative advantages and potential for improving ED admissions prediction. The discussion concludes with an overview of limitations and future directions for advancing predictive modeling in emergency medicine in Section 7.5.

7.1 Data requirements and format differences

It was observed that the standalone ML model employs each row to encompass all tests conducted at a single timestamp per patient. This allows the model to consider combinations of laboratory tests associated with ED admissions, thereby capturing nuanced relationships crucial for accurate predictions. Similarly, the PPM model with event-level features represents each laboratory test individually but retains the temporal sequence of tests, which results in comparable performance to the ML model. On the other hand, the PPM model with case-level features condenses all test results into a single row per patient, potentially missing the critical temporal dynamics necessary for precise forecasting. This comparison underscores that both the standalone ML model and the event-level PPM model effectively leverage temporal and combined test information, while the case-level approach does not fully capture the dynamic nature of the data.

7.2 Learning curves

An analysis of learning curves reveals slight differences in performance between the PPM techniques and standalone ML model (see Figure 5.1). The standalone ML model demonstrates a steady decrease in both training and validation losses from around 0.70, with minimal fluctuations and losses stabilising near zero by 300 epochs. This close alignment suggests effective training with minimal overfitting or underfitting. In contrast, the PPM model with event-level features also starts with losses around 0.70 and shows a rapid decline in the first 50 epochs, stabilising near zero by around 250 epochs. However, this model exhibits minor fluctuations in the training loss curve after 50 epochs, indicating slightly less stability. The PPM model with case-level features starts with losses around 0.68 and shows a steady decrease in training loss, but the validation loss decreases more slowly and eventually flattens out, leading to a widening gap between the two over 300 epochs. This growing divergence suggests potential overfitting, as the model may be learning the training data too well but struggles to generalise as effectively to the validation set. While both the ML and PPM event-level models effectively learn and converge, the standalone ML model is notably more robust and stable, with smoother performance and better generalisation compared to both the PPM models.

7.3 Predictive performance

Predictive performance analysis consistently favours the standalone ML and PPM models with event-level features over the PPM model with case-level features in the accuracy, precision and recall metrics (see Table 5.1). The standalone ML and PPM with event-level features models achieve near-perfect classification accuracy, reflecting robustness in distinguishing between ED and non-ED cases. Overall, the standalone ML model demonstrates performance comparable to the PPM model with event-level features, with minor metric differences ranging from 0.3% to 0.9%, primarily due to the standalone ML model having one more false positive. In contrast, the case-level PPM model exhibits lower performance levels, facing challenges such as misclassification and difficulty in handling class imbalances.

7.4 Classifying ED admissions with ML and PPM

The findings demonstrate that both the standalone ML model and PPM model with event-level features achieve similar performance in predicting ED admissions. Standalone ML model excels by leveraging complex data relationships, processing aggregated test data at specific timestamps and capturing the combined effects of multiple tests. Similarly, PPM model with event-level features effectively tracks individual test results while preserving temporal information, leading to comparable predictive accuracy.

While the event-level PPM model and standalone ML model offer valuable insights into patient processes over time, the PPM model with case-level features struggles with capturing temporal changes and addressing class imbalances. Both approaches, especially when enhanced with advanced techniques and comprehensive datasets, show considerable potential for improving predictions related to ED overcrowding and resource management.

Overall, the standalone ML model's performance is comparable to the PPM model with event-level features, with the main difference being a single additional false positive in the ML model. In a medical context, this single additional false positive is relatively inconsequential, as it is generally preferable to be cautious by admitting a patient who might not need urgent care rather than risking a missed critical case.

However, the standalone ML model significantly outperforms the PPM model with case-level features in terms of accuracy, precision and recall. This suggests that while PPM with event-level features can achieve similar performance to traditional ML methods, PPM with case-level features is less effective.

7.5 Limitations and future work

The study faces several limitations. Primarily, the dataset includes only hospital admission data, excluding information from general practitioner visits and pre-hospitalisation tests, which limits the model's ability to capture a full patient medical history. Future research could integrate additional clinical variables such as demographic information, coexisting medical conditions and severity scores to provide a more comprehensive patient profile for prediction. This expansion would enable models, including those based on DL architectures, to capture a broader range of factors influencing ED admissions and improve overall predictive accuracy. Furthermore, acquiring a dataset that includes information from general practitioner visits and pre-hospitalisation tests would enrich the model's understanding of patient health trajectories. This comprehensive dataset could potentially enhance the models' ability to predict ED admissions with greater precision and reliability. Therefore, future efforts should focus on acquiring and integrating such comprehensive datasets to advance predictive modelling in emergency medicine effectively.

Another limitation is the inherent class imbalance in the dataset, with 60% of patients admitted to the ED and 40% not admitted. This imbalance complicates model training and generalisation, particularly impacting the PPM model with case-level features. The research highlights the superior performance of standalone ML and event-level PPM models in predictive tasks and suggests that future work should focus on refining PPM techniques, especially those involving case-level features. Exploring advanced DL techniques or ensemble methods could offer performance improvements by better capturing complex temporal and sequential dependencies in clinical data.

To address class imbalance, future research could explore strategies such as using generative adversarial networks (GANs) to generate synthetic data, which might help balance the dataset and improve model robustness. Additionally, mitigating overfitting in the PPM model with case-level features could involve techniques such as regularisation or increasing the size of the validation set. Further tuning of hyper-parameters, like the learning rate or model architecture, might also help in reducing overfitting and improving generalisation. Conducting external validation studies across diverse healthcare settings and populations is essential to assess the generalisability and effectiveness of the models in real-world clinical environments. Future work should also focus on both model interpretability and explainability, as these aspects are crucial for establishing understanding and trust within the medical domain.

Bibliography

[1] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. Machine learning from theory to algorithms: An overview. *Journal of Physics: Conference Series*, 1142(1):012012, 2018. doi:10.1088/1742-6596/1142/1/012012.

[2] Qi An, Saifur Rahman, Jingwen Zhou, and James Jin Kang. A comprehensive review on machine learning in healthcare industry: Classification, restrictions, opportunities and challenges. *Sensors*, 23(9):4178, 2023. doi:10.3390/s23094178.

[3] Mohammad Mahdi Bejani and Mehdi Ghatee. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 54(8):6391–6438, 2021. doi:10.1007/s10462-021-09975-1.

[4] Adrian Boyle, Kathleen Beniuk, Ian Higginson, and Paul Atkinson. Emergency department crowding: Time for interventions and policy evaluations. *Emergency medicine international*, 2012:838610, 2012. doi:10.1155/2012/838610.

[5] Justin Boyle, Melanie Jessup, Julia Crilly, David Green, James Lind, Marianne Wallis, Peter Miller, and Gerard Fitzgerald. Predicting emergency department admissions. *Emergency Medicine Journal*, 29(5):358–365, 2012. doi:10.1136/emj.2010.103531.

[6] M. H. B. Carmalt, P Freeman, A. J. H. Stephens, and Thomas Patterson Whitehead. Value of routine multiple blood tests in patients attending the general practitioner. *British Medical Journal*, 1(5696):620–623, 1970. doi:10.1136/bmj.1.5696.620.

[7] Marcelo Dallagassa, Cleiton Garcia, Edson Scalabrin, Sergio Ioshii, and Deborah Carvalho. Opportunities and challenges for applying process mining in healthcare: a systematic mapping study. *Journal of Ambient Intelligence and Humanized Computing*, 13(4):165–182, 2021. doi:10.1007/s12652-021-02894-7.

[8] Johannes De Smedt and Jochen De Weerdt. Predictive process model monitoring using long short-term memory networks. *Engineering Applications of Artificial Intelligence*, 133(12):108295, 2024. doi:10.1016/j.engappai.2024.108295.

[9] Chiara Di Francescomarino and Chiara Ghidini. *Predictive Process Monitoring*, pages 320–346. Springer International Publishing, 2022. doi:10.1007/978-3-031-08848-3_10.

[10] Mahidhar Reddy Dwarampudi and Subba Reddy. Effects of padding on lstms and cnns. arXiv, 2019. doi:10.48550/arXiv.1903.07288.

[11] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger Mark, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi:10.1161/01.cir.101.23.e215.

[12] Hannah Harrison. Navigating critical care: Anesthesia and surgical considerations in emergency situations. Zenodo, 2023. doi:10.5281/zenodo.8361832.

[13] Azriel Henry, Sunil Gautam, Samrat Khanna, Khaled Rabie, Thokozani Shongwe, Pronaya Bhattacharya, Bhisham Sharma, and Subrata Chowdhury. Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors*, 23(2):1–22, 2023. doi:10.3390/s23020890.

[14] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5(2):01–11, 2015. doi:10.5121/ijdkp.2015.5201.

[15] Tim Hucker, G Mitchell, L Blake, E Cheek, V Bewick, M Grocutt, Lui Forni, and R Venn. Identifying the sick: Can biochemical measurements be used to aid decision making on presentation to the accident and emergency department. *British journal of anaesthesia*, 94(06):735–741, 2005. doi:10.1093/bja/aei122.

[16] Khalid Ijaz, Zawar Hussain, Jameel Ahmad, Syed Farooq Ali, Muhammad Adnan, and Ikramullah Khosa. A novel temporal feature selection based lstm model for electrical short-term load forecasting. *IEEE Access*, 10:82596–82613, 2022. doi:10.1109/ACCESS.2022.3196476.

[17] Kazuki Irie, Zoltán Tüske, Tamer Alkhouli, Ralf Schlüter, and Hermann Ney. Lstm, gru, highway and a bit of attention: An empirical overview for language modeling in speech recognition. In *Interspeech*, pages 3519–3523, 2016. doi:10.21437/Interspeech.2016-491.

[18] Mieke Jans, Michael Alles, and Miklos Vasarhelyi. Process mining of event logs in auditing: Opportunities and challenges. *SSRN Electronic Journal*, pages 1–32, 2010. doi:10.2139/ssrn.1578912.

[19] Stuart Jarvis, Caroline Kovacs, Tessy Badriyah, Jim Briggs, Mohammed Mohammed, Paul Meredith, Paul Schmidt, Peter Featherstone, David Prytherch, and Gary Smith. Development and validation of a decision tree early warning score based on routine laboratory test results for the discrimination of hospital mortality in emergency medical admissions. *Resuscitation*, 84(11):1494–1499, 2013. doi:10.1016/j.resuscitation.2013.05.018.

[20] Alistair Johnson, Lorenzo Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV (version 3.0), 2024. doi:10.13026/hxp0-hg59.

[21] Alistair E. W. Johnson, Lorenzo Bulgarelli, Linda Shen, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023. doi:10.1038/s41597-022-01899-x.

[22] Patricia Kipnis, Benjamin J. Turk, David A. Wulf, Juan Carlos LaGuardia, Vincent Liu, Matthew M. Churpek, Santiago Romero-Brufau, and Gabriel J. Escobar. Development and validation of an electronic medical record-based alert score for detection of inpatient deterioration outside the icu. *Journal of Biomedical Informatics*, 64(5):10–19, 2016. doi:10.1016/j.jbi.2016.09.013.

[23] Rodney Kizito, Phillip Scruggs, Xueping Li, Michael Devinney, Joseph Jansen, and Reid Kress. Long short-term memory networks for facility infrastructure failure and remaining useful life prediction. *IEEE Access*, 9:67585–67594, 2021. doi:10.1109/ACCESS.2021.3077192.

[24] Anders Krogh. What are artificial neural networks? *Nature Biotechnology*, 26(2):195–197, 2008. doi:10.1038/nbt1386.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi:10.1038/nature14539.

[26] Suja Cherukullapurath Mana, G. Kalaiarasi, R. Yogitha, L Suji Helen, and R. Senthamil Selvi. Application of machine learning in healthcare: An analysis. pages 1611–1615, 2022. doi:10.1109/ICESC54411.2022.9885296.

[27] Niels Martin, Nils Wittig, and Jorge Munoz-Gama. *Using Process Mining in Healthcare*, pages 416–444. Springer International Publishing, 2022. doi:10.1007/978-3-031-08848-3_14.

[28] Maad M. Mijwel. Artificial neural networks advantages and disadvantages. *Mesopotamian Journal of Big Data*, 2021:29–31, 2021. doi:10.58496/MJBD/2021/006.

[29] Mohammed Mohammed, Gavin Rudge, Duncan Watson, Gordon Wood, Gary Smith, David Prytherch, Alan Girling, and Andrew Stevens. Index blood tests and national early warning scores within 24 hours of emergency admission can predict the risk of in-hospital mortality: A model development and validation study. *PLOS ONE*, 8(5):e64340, 2013. doi:10.1371/journal.pone.0064340.

[30] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. *Fundamentals of Artificial Neural Networks and Deep Learning*, pages 379–425. Springer International Publishing, 2022. doi:10.1007/978-3-030-89010-0_10.

[31] Bossy Mostafa, Noha El-Attar, Samy Abd-Elhafeez, and Wael Awad. Machine and deep learning approaches in genome: Review article. *Alfarama Journal of Basic Applied Sciences*, 2(1):105–113, 2021. doi:10.21608/ajbas.2020.34160.1023.

[32] Gábor Petneházi. Recurrent neural networks for time series forecasting. arXiv, 2019. doi:10.48550/arXiv.1901.00069.

[33] D.R. Prytherch, J.S. Sirl, P. Schmidt, P.I. Featherstone, P.C. Weaver, and G.B. Smith. The use of routine laboratory data to predict in-hospital death in medical admissions. *Resuscitation*, 66(2):203–207, 2005. doi:10.1016/j.resuscitation.2005.02.011.

[34] Fazla Rabbi, Debapriya Banik, Niamat Ullah Ibne Hossain, and Alexandr Sokolov. Using process mining algorithms for process improvement in healthcare. *Healthcare Analytics*, 5:100305, 2024. doi:10.1016/j.health.2024.100305.

[35] L.B. Roberts. The normal ranges, with statistical analysis for seventeen blood constituents. *Clinica Chimica Acta*, 16(1):69–78, 1967. doi:10.1016/0009-8981(67)90271-9.

[36] Eric Rojas, Marcos Sepúlveda, Jorge Munoz-Gama, Daniel Capurro, Vicente Traver, and Carlos Fernández-Llatas. Question-driven methodology for analyzing emergency

room processes using process mining. *Applied Sciences*, 7(3):302, 2017. doi:10.3390/APP7030302.

[37] Dilip Sharma, Dhruva Chakravarthi, Raja Boddu, Ah Ad, Maruthi Ayyagari, and Dr-Md Mohiddin. *Effectiveness of Machine Learning Technology in Detecting Patterns of Certain Diseases Within Patient Electronic Healthcare Records*, pages 73–81. 2023. doi:10.1007/978-981-19-0108-9_8.

[38] Omkar Shinde, Rishikesh Gawde, and Anurag Paradkar. Image caption generation methodologies. *International Research Journal of Engineering and Technology (IRJET)*, 08(04):3961–3966, 2021.

[39] Adam J. Singer, Peter Viccellio, Henry C. Thode Jr., Jay L. Bock, and Mark C. Henry. Introduction of a stat laboratory reduces emergency department length of stay, 2008. doi:10.1111/j.1553-2712.2008.00065.x.

[40] Yayat Sujatna, Adhitio Satyo, Widi Hastomo, Nia Yuningsih, Dody Arif, Sri Handayani, Aqwam Rosadi Kardian, Ire Wardhani, and L.M Rere. Stacked lstm-gru long-term forecasting model for indonesian islamic banks. *Knowledge Engineering and Data Science*, 6(02):215, 2023. doi:10.17977/um018v6i22023p215-250.

[41] Abeer Thamara, Mohamed Elsersy, Ahmed Sherif, Hossam Hassan, Omar Abdelsalam, and Khaled H. Almotairi. A novel classification of machine learning applications in healthcare. In *2021 3rd IEEE Middle East and North Africa COMMunications Conference, MENACOMM 2021*, pages 80–85, 2021. doi:10.1109/MENACOMM50742.2021.9678232.

[42] Jack V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11):1225–1231, 1996. doi:10.1016/S0895-4356(96)00002-9.

[43] Dennis J. Van De Wijngaart, Jolanda Scherrenburg, Lisette Van Den Broek, Nadine Van Dijk, and Pim M.W. Janssens. A survey of doctors reveals that few laboratory tests are of primary importance at the emergency department. *Diagnosis*, 1(3):239–244, 2014. doi:10.1515/dx-2014-0025.

[44] Wil Van Der Aalst. Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, 3(2):1–17, 2012. doi:10.1145/2229156.2229157.

[45] Wil van der Aalst. *Data Science in Action*, pages 3–23. Springer, 2016. doi:10.1007/978-3-662-49851-4_1.

[46] Marc van Wijk, Arthur Bohnen, and Johan Lei. Analysis of the practice guidelines of the dutch college of general practitioners with respect to the use of blood tests. *Journal of the American Medical Informatics Association : JAMIA*, 6(4):322–331, 1999. doi:10.1136/jamia.1999.0060322.

[47] Bemali Wickramanayake, Chun Ouyang, Yue Xu, and Catarina Moreira. Generating multi-level explanations for process outcome predictions. *Engineering Applications of Artificial Intelligence*, 125(5):106678, 2023. doi:10.1016/j.engappai.2023.106678.

[48] R. Wirth. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pages 29–40, 2000.

[49] Kun-Hsing Yu, Andrew Beam, and Isaac Kohane. Artificial intelligence in healthcare. *Nature Biomedical Engineering*, 2(10):719–731, 2018. doi:10.1038/s41551-018-0305-z.

Appendix A

Laboratory tests and descriptions

Table A.1: Laboratory tests and their corresponding descriptions

Laboratory Tests	**Description**
Alanine Aminotransferase (ALT)	ALT is an enzyme predominantly found in the liver. An ALT blood test is used to evaluate liver health as elevated ALT levels in the blood may indicate liver damage or a liver condition.
Albumin	An albumin blood test measures protein levels in blood plasma. Low levels can indicate kidney or liver disease, inflammation or infections, while high levels may result from dehydration or severe diarrhea.
Aspartate Aminotransferase (AST)	AST is an enzyme present in various tissues like the liver, heart, pancreas and muscles. An AST blood test aids healthcare providers in evaluating liver function.
Bands	Also called band neutrophils, these are immature white blood cells produced in response to infection or inflammation. They are identified in a blood test to assess the body's immune response, where high levels indicate increased production to fight infection or inflammation.
Bilirubin, Total	A bilirubin test measures the level of bilirubin in the blood, a yellow pigment found in bile. Elevated levels may indicate liver dysfunction or blocked bile ducts.
C-Reactive Protein	A c-reactive protein (CRP) test measures the level of CRP in the blood, which is released by the liver in response to inflammation. This test helps diagnose and monitor conditions like infections and autoimmune diseases.
Calcium, Total	Calcium, an essential mineral, is monitored through blood tests to ensure healthy levels. Abnormal calcium levels can signal medical conditions.

Cholesterol, HDL	High-density lipoprotein (HDL) cholesterol, known as "good cholesterol," aids in removing excess cholesterol by transporting it to the liver for elimination through feces. Adequate levels of HDL help prevent artery plaque buildup, lowering the risk of heart disease and stroke.
Cholesterol, LDL, Measured	Low-density lipoprotein (LDL), a type of lipoprotein in the blood, carries cholesterol and fats throughout the bloodstream. High levels of LDL, often referred to as "bad cholesterol," increase the risk of stroke and heart disease due to their cholesterol-rich composition.
Cholesterol, Total	Total cholesterol measures the combined amount of cholesterol in the blood, encompassing both LDL and HDL. This total value helps gauge the risk of heart disease.
Creatine Kinase (CK)	CK is an enzyme present in skeletal muscle, heart muscle and brain. Increased CK levels in the bloodstream can signal damage or disease in these tissues.
Creatinine	The creatinine test assesses kidney function by measuring creatinine levels in the blood. Creatinine, a waste product filtered by the kidneys, may indicate kidney disease if levels are abnormal.
Glucose	A blood glucose test measures the amount of sugar in the blood, commonly used to screen for Type 2 diabetes. It can be done through a finger prick or a blood draw from a vein.
Granulocyte Count	Granulocytes, the most prevalent white blood cells, release enzymes from their granules to combat immune system threats like infections, allergies or asthma. Produced in the bone marrow from stem cells, granulocytes have a short lifespan of a few days.
Hematocrit	A hematocrit test measures the percentage of red blood cells in the blood, crucial for oxygen transport. Abnormal results can indicate blood disorders or other medical issues.
Hemoglobin	A hemoglobin test measures the levels of hemoglobin, the primary component of red blood cells. It is primarily used to detect anemia.
Lymphocytes	Lymphocytes are white blood cells that help fight cancer and infections. Their levels can be measured in a routine blood test and may vary based on factors like age, race and lifestyle.
NTproBNP	N-terminal pro b-type natriuretic peptide (NT-proBNP) is a protein used to make the BNP hormone. Like BNP, the heart makes larger amounts of NT-proBNP when it has to work harder to pump blood, potentially indicating heart failure.

Platelet Count	A platelet count measures the number of platelets in the blood, which are cells that help with clotting. Low platelet levels can indicate cancer, infections or other health issues, while high levels can increase the risk of blood clots or stroke.
Potassium	A potassium blood test measures the amount of potassium, an electrolyte, in the blood. Potassium is essential for proper cell, nerve, heart and muscle function. Abnormal potassium levels can indicate medical problems.
Protein, Total	A total protein blood test measures the amount of all proteins in the blood, primarily albumin and globulin. This test helps assess the overall health and can indicate issues with the liver, kidneys or other conditions affecting protein levels in the body.
Sodium	A sodium blood test measures the amount of sodium, an essential electrolyte, in the blood. Sodium helps regulate fluid balance, pH levels and nerve and muscle function. Abnormal sodium levels can indicate kidney issues, dehydration or other medical conditions.
Thyroid Stimulating Hormone	Thyroid-stimulating hormone (TSH) prompts the thyroid to release hormones affecting metabolism. High TSH typically indicates hypothyroidism, while low TSH indicates hyperthyroidism.
Thyroxine (T4), Free	A T4 test diagnoses thyroid conditions by measuring the thyroid hormone T4. Abnormal levels suggest thyroid issues.
Triglycerides	Triglycerides are common fats found in food and stored in the body. High levels increase the risk of heart attacks and strokes.
Troponin T	A troponin T blood test measures the levels of troponin T, a protein released into the bloodstream during a heart attack or heart muscle damage. Elevated levels of troponin T indicate heart injury and help diagnose conditions such as myocardial infarction.
Urea Nitrogen	The blood urea nitrogen test measures urea nitrogen in the blood to assess kidney function. Abnormal levels can indicate kidney damage or other health conditions.
White Blood Cells	White blood cells are part of the immune system, protecting the body from infection by attacking unknown organisms that enter the bloodstream and tissues.