

Titre: Toward Improved ICU Care – Predicting Blood Draw Frequency at Admission Time with Deep Learning

Auteur: Rola Al-Mallah

Date: 2025

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Al-Mallah, R. (2025). Toward Improved ICU Care – Predicting Blood Draw Frequency at Admission Time with Deep Learning [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/62550/>

 **Document en libre accès dans PolyPublie**

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/62550/>

Directeurs de recherche: Alejandro Quintero

Programme: Génie informatique

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Toward Improved ICU Care – Predicting Blood Draw Frequency at
Admission Time with Deep Learning**

ROLA AL-MALLAH

Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie informatique

Janvier 2025

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Toward Improved ICU Care – Predicting Blood Draw Frequency at
Admission Time with Deep Learning**

présenté par **Rola AL-MALLAH**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Samuel PIERRE, président

Alejandro QUINTERO, membre et directeur de recherche

Martine BELLAÏCHE, membre

DEDICATION

This thesis is dedicated to the most significant influences in my life.

To my mother and father, whose steadfast love and unwavering support have profoundly shaped every step of this journey. Without you, my achievements would be but a fraction of what they are today. Your faith in me—often stronger than my own—has been my guiding light, illuminating the path forward and inspiring me to reach heights I once thought unattainable.

To my siblings, whose camaraderie and encouragement have provided me with strength and joy. Your laughter and support have been essential, reminding me that I am never alone on this journey. Thank you for believing in me and for always cheering me on.

To my husband, with whom I have shared countless years of love and partnership. Your steadfast support has turned every challenge into an opportunity for growth. You inspire me daily to reach new heights, and I deeply admire your wisdom, strength, and grace in leading our family and helping me achieve my goals. Your encouragement ignites a passion within me, fueling my confidence and ambition. In your embrace, I find not just a partner, but a wellspring of inspiration that propels me to be the best version of myself.

To my beautiful daughters, who have shown remarkable patience and understanding as I pursued my goals. Your joy in my work and studies has been a constant reminder of what truly matters. You have instilled in me a renewed sense of purpose and determination, inspiring me to strive for excellence every day. You are the brightest lights in my life, and without your love and support, none of this would have been possible. Thank you for being my motivation and my greatest treasures.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my supervisor, Dr. Alejandro Quintero, whose invaluable guidance and support made this research possible. Your expertise and encouragement have been instrumental in shaping my work and providing the resources and funding necessary for my research.

I extend my thanks to a special mentor for her insightful feedback and constructive criticism throughout the process. These professional exchanges helped me complete this thesis with confidence and clarity, enabling me to refine my ideas and strengthen my arguments.

Lastly, I wish to thank my family and friends for their unwavering support and understanding during this journey. Your belief in me has been a constant source of motivation, and your encouragement has carried me through the toughest times.

This research is dedicated to all who have contributed to my growth as a researcher and individual. Thank you for being part of this journey.

RÉSUMÉ

L'introduction de l'intelligence artificielle (IA) dans le domaine médical offre de nouvelles perspectives pour améliorer la qualité des soins et optimiser les pratiques cliniques. Cette technologie devient de plus en plus essentielle en particulier dans des environnements critiques tels que les unités de soins intensifs (USI). En combinant la puissance de l'apprentissage automatique et de l'analyse prédictive, les professionnels de la santé peuvent prendre des décisions plus éclairées, améliorant ainsi les résultats pour les patients. L'une des principales applications de cette avancée technologique concerne la gestion des prélèvements sanguins, où l'IA peut transformer la façon dont la fréquence des prises de sang, connue sous le nom de phlébotomie, est gérée. Les techniques de gestion de phlébotomie ont été un domaine de recherche continu dans les hôpitaux, en particulier dans les USI. Les progrès des protocoles et de la technologie ont conduit à des améliorations significatives ; cependant, le défi des prélèvements sanguins fréquents persiste, car ces tests influencent environ 70 % des décisions cliniques. Bien que certaines études aient utilisé l'apprentissage automatique pour prédire les résultats des tests sanguins juste avant la prise de décision, cette thèse se concentre sur la prévision de la fréquence des prélèvements sanguins dès le moment de l'admission en USI, en proposant un calendrier personnalisé des prises de sang tout au long du séjour du patient.

Cette thèse utilise des réseaux de neurones à mémoire longue et courte, un type spécialisé de réseau de neurones dans le domaine de l'IA, pour modéliser une solution visant à prédire la fréquence des prélèvements sanguins. Ce type de réseau de neurones est particulièrement adapté pour traiter des données séquentielles ou des séries temporelles, permettant de capturer des motifs temporels complexes et des dépendances. En considérant le problème comme une transition de la prédiction de séquences statiques à variables à l'aide de données temporelles, l'étude aborde les complexités des multi-sorties variables. Elle comprend une revue approfondie des techniques existantes visant à réduire les prélèvements sanguins et identifie les caractéristiques clés essentielles pour prédire la fréquence de phlébotomie par le biais de techniques de sélection des variables d'entrée et de sortie. Ce processus simplifie l'exploration des combinaisons de variables potentielles en évaluant les dépendances et en réalisant une analyse de corrélation pour identifier la multi colinéarité. De plus, la thèse examine l'application de la fonction de perte de correntropy utilisant un modèle avancé d'IA sur des ensembles de données médicales. Pendant l'entraînement, cette fonction de perte peut améliorer la précision en se concentrant sur les similitudes plutôt que sur une comparaison directe

entre la prédiction et la valeur réelle. Cela permet également de rendre l'entraînement moins sensible aux données non gaussiennes. Vu la nouveauté de ce domaine de recherche, un modèle de régression linéaire à multiples sorties a été entraîné pour établir une référence, ce qui a permis de développer les variables d'entrée appropriées pour le modèle avancé. Les variables temporelles et quantitatives ont été intégrées, ainsi que les signes vitaux, les résultats des tests sanguins, les données cliniques et les informations démographiques, afin d'améliorer la capacité du réseau de neurones à capturer les informations pertinentes. Grâce à ce cadre, nous avons réussi à prédire la fréquence des prises de sang avec une précision de $88,26\% \pm 3$ points (représentant la longueur de la séquence) et une erreur absolue moyenne de $\pm 0,96$ heure sur les valeurs de cette séquence.

Cette thèse jette les bases de la prédiction précise de la fréquence des prélèvements sanguins en USI dès le moment de l'admission, marquant ainsi une avancée significative dans les pratiques de gestion des phlébotomies grâce à l'application des modèles les plus avancés de l'IA.

ABSTRACT

Advancements in medicine driven by artificial intelligence (AI) are revolutionizing patient care, particularly in critical environments like Intensive Care Units (ICUs). By connecting the power of machine learning and predictive analytics, healthcare professionals can make more informed decisions, ultimately improving patient outcomes. One of the key areas that could be benefiting from this technological leap is blood management, where using AI can reshape how blood draw, known as phlebotomy, frequency is handled. Blood management techniques have been an ongoing area of research in hospitals, particularly in ICUs. Advances in protocols and technology have led to significant improvements; however, the challenge of frequent blood draws persists, as these tests inform approximately 70% of clinical decisions. While some studies have employed machine learning to predict blood test outcomes shortly before decisions are made, this thesis shifts the focus to forecasting blood draw frequency from the moment of ICU admission, presenting a tailored schedule of phlebotomy throughout the patient's stay.

This thesis employs Long-Short-Term-Memory (LSTM) networks, a specialized type of neural network (NN) within the realm of AI, to model a solution for optimizing blood draw timing. LSTMs are particularly well-suited for handling sequential data or time series data, allowing them to capture complex temporal patterns and dependencies. By treating the problem as a transition from static to variable sequence prediction, the study addresses the complexities of variable multi-output timestamps. It includes a thorough review of existing techniques aimed at reducing blood sampling and identifies key features essential for predicting phlebotomy frequency through feature selection techniques. This process streamlines the exploration of potential feature combinations by assessing dependencies and conducting correlation analysis to identify multicollinearity. Additionally, the thesis examines the application of Correntropy Loss on medical datasets. During training, this loss has the potential to improve accuracy as it focusses on similarities and is less affected by data that are not Gaussian. Given the novelty of this research area, a multi-output linear regression model was trained to establish a baseline, which informed the development of appropriate input features for the LSTM model. Temporal and quantitative variables were integrated alongside vital signs, blood test results, clinical data, and demographic information to enhance the LSTM's capability to capture relevant information. With this framework, we achieved the ability to predict multi-output timestamps with an accuracy of $88.26\% \pm 3$ timestamps (representing blood draw timings) and an average mean absolute error (MAE) of ± 0.96 hours.

This thesis lays the groundwork for accurately predicting blood sample frequency in the ICU at the time of admission, marking a significant advancement in blood management practices through the application of AI's most advanced models.

TABLE OF CONTENTS

DEDICATION	III
ACKNOWLEDGEMENTS	IV
RÉSUMÉ	V
ABSTRACT.....	VII
LIST OF TABLES	XII
LIST OF FIGURES.....	XIII
LIST OF SYMBOLS AND ABBREVIATIONS	XVI
LIST OF APPENDICES.....	XVII
CHAPTER 1 INTRODUCTION.....	1
1.1 Problem Statement.....	2
1.2 Research Objectives	3
1.3 Scope	4
1.4 Thesis Outline	4
PART I LITERATURE REVIEW	6
CHAPTER 2 ICU PATIENT MONITORING.....	7
2.1 Introduction to Patient Monitoring in the ICU	7
2.1.1 Role of Blood Sampling in the Management of ICU Patients.....	8
2.1.2 Modern blood sampling techniques and practices	8
2.1.3 Influence of AI in ICU Patient Management.....	11
2.2 Medical Patient Databases.....	15
2.2.1 Review of Key Database Types and Their Applications.....	15
2.2.2 Exploiting MIMIC for Progress in Critical Care Research	17
2.3 Understanding Multi-Output Regression	18

2.4	LSTM Theory	24
2.4.1	Introduction to LSTMs	24
2.4.2	Understanding the Structure of LSTMs.....	25
2.4.3	Impact of LSTMs on Multi-Output Regression Performance	29
2.5	Evaluation Metrics and Loss functions	31
2.5.1	Key Metrics for Multi-Output Regression.....	31
2.5.2	Loss Functions and Their Influence on Inference	33
2.6	Conclusion	34
PART II	CONTRIBUTION.....	36
CHAPTER 3	METHODOLOGY	37
3.1	Dataset Creation.....	37
3.1.1	Feature Definition and Units.....	38
3.1.2	Source and Methods for Feature and Target.....	40
3.1.3	Data preprocessing	44
3.2	Model Configuration and Training	48
3.3	Tools and Technologies.....	49
3.4	Conclusion	49
CHAPTER 4	RESULTS	51
4.1	Data Preparation Outcomes	51
4.1.1	Results of Data Extraction from MIMIC-IV	51
4.1.2	Outcomes of Data Preprocessing	55
4.2	Model Training Results	64
4.2.1	Iterative training Process	67
4.2.2	Analysis of Loss Function Performance.....	73

4.3	LSTM Improvement through Hyperparameter Tuning.....	75
CHAPTER 5	DISCUSSION	84
5.1	Overview of the findings.....	84
5.1.1	Evaluation of Key Features.....	84
5.1.2	MSE vs Correntropy Loss	86
5.2	Limitations.....	87
5.3	Suggestions for Future Research	88
CHAPTER 6	CONCLUSION AND RECOMMENDATIONS.....	90
REFERENCES.....		91
APPENDICES.....		95

LIST OF TABLES

Table 2-1: Comparison of various strategies for reducing blood loss during blood sampling in ICU	10
Table 2-2: Overview of multioutput regression techniques and their applications.....	22
Table 2-3: LSTM techniques for variable multi-output regression.....	30
Table 2-4: Evaluation metrics for multi-output regression.....	32
Table 3-1: Patient features and data identifiers in MIMIC-IV.....	39
Table 4-1: Example of recorded timings for blood samples collected between patient admission and discharge from an ICU ward.....	54
Table 4-2: Correlation table of the significant relationship	59
Table 4-3: Top 20 and bottom 20 features chosen by each feature selection method	61
Table 4-4: Performance metrics of various kernel size (σ)	74
Table 4-5: Comparison of MAE and RMSE for values and sequence lengths across models	81

LIST OF FIGURES

Figure 2-1: Multiple monitoring devices connected to a patient who suffered TBI	7
Figure 2-2: Timeline of Multi-Output Regression Models.....	22
Figure 2-3 : Classification of multioutput regression models.....	24
Figure 2-4: Complete LSTM architecture using patient input features $x_1, \dots, x_p$ and blood test frequencies $t_1, \dots, t_{\text{discharge}}$	27
Figure 3-1: Overview of the methodology process for building a machine learning model for predicting blood test frequencies in the ICU.....	37
Figure 3-2: Relational schema of the tables used to extract the features and target values, guiding the formulation of the SQL queries for data extraction	43
Figure 4-1: SQL query for laboratory tests values from the MIMIC-IV database, i.e. le.itemid = 50983 corresponds to sodium (Na).....	52
Figure 4-2: SQL query for vital signs, i.e. ce.itemid = 220045 corresponds to the first measure of the heart rate of the patient at admission time to an ICU ward	53
Figure 4-3: SQL Query to capture blood test times representing individual draw events	54
Figure 4-4: Number of patients categorized by blood draw frequency.....	56
Figure 4-5: Visualization of numerical features - a) before applying outlier detection Z-score and IQR - b) after applying Z-score and IQR.....	58
Figure 4-6: Correlation heatmap for numerical features including vital signs, blood test results and clinical patient data > 0.5	60
Figure 4-7: Venn diagramm of the – a) top 20 features and b) bottom 20 features – in different selection methods	63
Figure 4-8: Venn diagram of the top 50 – a) and bottom 50 features – b) in different selection methods	64
Figure 4-9: MultiOutput Linear Regression – a) using all features b) removed least significant features from Ensemble Method c) using <i>time_interval_t0</i> as additional feature generated	

better result, removing calcium, and chloride based on correlation and testing with or without the features	66
Figure 4-10: Single LSTM trained on all features - Scatter plot of the true vs. predicted sequence length using the red dotted line as benchmark	69
Figure 4-11: Single LSTM trained on all features - Distribution of timestamps comparing both the actual and predicted sequence lengths	69
Figure 4-12: Single LSTM trained on all features - Box plot of MAE and RMSE for the value in each timestamp alongside R^2	70
Figure 4-13: Single LSTM trained on all features – MAE and RMSE for each timestamp t_0 to t_{50} with the overall red dotted line.....	70
Figure 4-14: Single LSTM trained on selected features - Scatter plot of the true vs. predicted sequence length using the red dotted line as benchmark	71
Figure 4-15: Single LSTM trained on selected features - Distribution of timestamps comparing both the actual and predicted sequence lengths.....	71
Figure 4-16: Single LSTM trained on selected features - Box plot of MAE and RMSE for the value in each timestamp alongside R^2	72
Figure 4-17: Single LSTM trained on selected features – MAE and RMSE for each timestamp t_0 to t_{50} with the overall red dotted line	72
Figure 4-18: Trend of the predicted values at each timestamp using 5 samples from the test set	73
Figure 4-19: Visualization of the performance metrics across various kernel size (σ).....	75
Figure 4-20: MAE and RMSE for each timestamp t_0 to t_{50} with tuned LSTM using selected features, gender, time_interval_ t_0 and number of blood draw - MSE Loss.....	77
Figure 4-21: Box plot of MAE and RMSE for the value in each timestamp alongside R^2 with tuned LSTM using selected features, gender, time_interval_ t_0 and number of blood draw - MSE Loss.....	77
Figure 4-22: Distribution of timestamps of actual and predicted sequence with tuned LSTM using selected features, gender, time_interval_ t_0 and number of blood draw - MSE Loss	78

Figure 4-23: Scatter plot of the true vs. predicted sequence length with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - MSE Loss.....	78
Figure 4-24: Evaluation metrics with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - MSE Loss	79
Figure 4-25: MAE and RMSE for each timestamp t ₀ to t ₅₀ with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - Correntropy Loss	79
Figure 4-26: Box plot of MAE and RMSE for the value in each timestamp alongside R ² with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - Correntropy Loss	79
Figure 4-27: Distribution of timestamps of actual and predicted sequence with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - Correntropy Loss.	80
Figure 4-28: Scatter plot of the true vs. predicted sequence length with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood draw - Correntropy Loss.....	80
Figure 4-29: Evaluation metrics with tuned LSTM using selected features, gender, time_interval_t ₀ and number of blood - Correntropy Loss.....	81

LIST OF SYMBOLS AND ABBREVIATIONS

AI - Artificial Intelligence

CDSS - Clinical Decision Support System

CTE - Common Table Expression

EHR - Electronic Health Record

ICU - Intensive Care Unit

IQR - Interquartile Range

LSTM - Long Short Term Memory

MAE - Mean Absolute Error

ML - Machine Learning

MSE - Mean Square Error

MIMIC - Medical Information Mart for Intensive Care

NHANES - National Health and Nutrition Examination Survey

POCT - Point-Of-Care Testing

RNN - Recurrent Neural Network

RMSE - Root Mean Square Error

SL - Sequence Length

LIST OF APPENDICES

APPENDIX A Additional Tables.....	95
APPENDIX B Additional Figures	101
APPENDIX C Python Code	108

CHAPTER 1 INTRODUCTION

Advancements in medicine driven by artificial intelligence (AI) are revolutionizing patient care, particularly within hospital settings. AI serves as a critical tool for healthcare professionals, enabling informed decision-making, assisting in treatment strategies, and predicting patient outcomes. By leveraging historical patient data and clinical experiences, AI systems enhance diagnostic accuracy and facilitate the personalization of treatment plans, making them indispensable in modern medical practice. Recently, Intensive Care Units (ICUs) have been the focus of extensive research exploring how we can leverage the power of AI in this critical environment. The ICU is a section in a hospital that provides intensive care and treatment for patients who are critically ill or injured. The conditions treated in the ICU are diverse, ranging from severe infections, heart attacks, and strokes to major trauma, respiratory failure, and postoperative complications. These life-threatening conditions require vigilant monitoring and support. To meet these needs, ICUs are equipped with advanced medical technology and highly trained healthcare professionals working together to provide round-the-clock care. Thus, the primary goals of ICU care are to stabilize patients, prevent further deterioration, and facilitate recovery. In other words, ICU teams prioritize timely intervention and monitoring to address changes in patient conditions promptly.

Over the years, advancements in medical technology and the evolution of intensive care practices have helped ICU professionals to be very well-equipped to face the complexities of critical illness and injury. One of the advancements is blood sampling in other words, phlebotomy. It is a critical diagnostic tool in the healthcare system, essential for assessing a patient's condition, guiding medical interventions, and supporting approximately 60-70% of clinical decisions [1]. However, frequent blood sampling can lead to issues such as patient discomfort, increased risk of infection, and the need for careful management of blood volume to prevent iatrogenic anemia. Despite the remarkable progress in patient care, the potential for iatrogenic anemia persists, over 70% of patients admitted to the ICU will be anemic by their second day [2], [3], and almost 95% by their third day [4]. Iatrogenic anemia in the ICU is concerning because it can worsen existing medical conditions and contribute to complications that could increase mortality rates. Therefore, healthcare providers in the ICU must carefully balance the need for phlebotomy with the potential

risks of iatrogenic anemia. Strategies to minimize blood loss, such as using smaller blood collection tubes, implementing blood conservation techniques, and optimizing the timing of diagnostic procedures, are essential for mitigating the risk of iatrogenic anemia and maintaining optimal patient care in the ICU [5], [6]. Ultimately, the integration of AI in healthcare not only aims to improve patient outcomes but also seeks to empower healthcare professionals with tools that enhance their expertise and capabilities. As technology continues to evolve, the potential for AI to transform healthcare is vast, paving the way for a more effective and responsive healthcare system.

1.1 Problem Statement

Among the strategies to minimize blood loss, determining the appropriate frequency of blood samples presents a significant challenge to healthcare professionals. Factors such as the severity of the patient's condition, the progression of their illness, and the availability of resources make it difficult for healthcare professionals to establish standardized protocols for blood sampling, as highlighted by existing research [7]. Furthermore, the authors in [8] emphasize that a one-size-fits-all approach to the prediction of blood sample frequency may lead to inefficiencies and increased waste, noting that “*standardization may actually lead to more waste*” due to the diverse nature of surgical complexities and patient responses. Given these challenges, there is a critical need for a more personalized approach to blood sampling [9], [10]. Leveraging AI and particularly the deep learning models in Machine Learning (ML) could help in the prediction of the optimal timing for blood sampling based on patient-specific features at the time of ICU admission.

Existing research in this field assesses mainly the necessity and values of blood tests in ICU settings [11],[12],[13],[14]. These studies use deep learning techniques to predict blood test results at specific points in time, aiming to assess whether individual tests are necessary and valuable, particularly when there is uncertainty in decision-making or when frequent blood draws have already occurred. This approach helps determine whether a blood test will provide significant information, guiding whether to proceed with the test to minimize patient discomfort and complications. However, this method is limited by its lack of a broader temporal perspective, as it focuses on isolated points in time rather than considering the full sequence of testing over the patient's ICU stay.

In contrast, there is a need to shift focus towards predicting the entire sequence of blood samples from the point of ICU admission. This shift would involve developing models that forecast not only the number but also the timing of required blood samples during the patient's entire ICU stay. By providing this comprehensive information upfront, healthcare professionals can make more informed decisions about patients' diagnostic and monitoring needs throughout their ICU stay, potentially improving patient management and reducing unnecessary interventions. Addressing this gap could significantly reduce iatrogenic anemia, enhance patient monitoring techniques, improve resource allocation, and contribute to more effective blood preservation techniques. This study aims to fill this void by developing a predictive model to forecast blood sampling frequencies tailored to each patient admitted to the ICU.

1.2 Research Objectives

The main objective of this study is to build a predictive model for determining the frequency of blood sampling for ICU patients. We will be using Long-Short-Term-Memory (LSTM) networks which are particularly well-suited for sequence-to-sequence predictions due to their effectiveness in managing time series data. More specifically, we will focus on the following specific goals:

1. Design a comprehensive dataset that includes all relevant variables for predicting blood sampling frequency in an ICU setting. This will involve discovering or generating synthetic data by investigating the processes surrounding blood sample collection with the goal of accurately replicating the sequence of events and ensuring the data's realism, which is essential for enhancing the model's generalization to unseen data.
2. Analyse feature interaction for accurate predictions, including vital signs, blood test results, and patient demographics. This will involve a thorough literature review to determine which features influence blood sampling frequency and to highlight any novel features not previously reported.
3. Construct an LSTM architecture to model the dynamics of blood sampling frequency, allowing for the analysis of time-series data in relation to patient characteristics. This involves defining the network layers and hyperparameters and implementing mechanisms to handle variable-length sequences effectively. Additionally, it will include training the model on the prepared dataset and fine-tuning it to optimize performance for accurate predictions.

4. Evaluate the performance of the LSTM model by examining its ability to handle static patient-specific input features and predict variable-length sequences of timing intervals. This evaluation will include the comparison of two loss functions - Mean Squared Error (MSE) and Correntropy Loss - and the use of a hybrid metric that accounts for both the length and values of the sequence to accurately measure the model's performance.

In summary, the findings will be documented to provide a detailed account of the model's effectiveness, the relevance of selected features, and the suitability of evaluation metrics and loss functions for the implementation of an ML model on the task of predicting blood sampling frequency in ICU settings. This study pioneers the use of predictive modeling for blood sampling frequency from the point of ICU admission.

1.3 Scope

This study focuses on developing and evaluating a predictive model for the task of estimating the frequency of blood sampling in ICU patients, leveraging machine learning's LSTM architecture. The research will encompass the identification and selection of relevant features from ICU patient data, the development and training of the LSTM model, and a comparative analysis with a baseline multi-output linear regression model. The scope includes evaluating various architecture designs, parameter choices and loss functions such as hyperparameters to determine their impact on model performance and selecting appropriate evaluation metrics to assess prediction performance. The study is limited to the context of adult ICU patients (> 18 years old) and does not extend to other medical settings or types of predictive models beyond the chosen LSTM and baseline approaches. Additionally, the analysis will be constrained to the dataset used for training and testing, and the findings will be specific to the blood sampling frequency prediction task. This focused approach ensures a thorough examination of the LSTM model's suitability for this task and provides insights into feature importance and model evaluation techniques thus generalization is left for future work.

1.4 Thesis Outline

This thesis is structured to provide a comprehensive exploration of the development of a predictive model to address the challenge of predicting blood sampling frequencies in the ICU. Following

this introduction, the subsequent chapters will delve into the existing literature related to ICU patient monitoring more specifically connected to blood sample reduction strategies and discuss multi-output regression in ML. The methodology section will detail the process of data collection and preprocessing, including the extraction of relevant features from a medical database. Furthermore, it will explain the architecture, training methodology, and evaluation metrics employed. The results and discussion section will present the findings of the study, analyze their implications, and discuss potential limitations. Finally, the conclusion and future work chapter will summarize the key findings and suggest opportunities for further research. Through this structured approach, this thesis aims to contribute to the growing body of knowledge on simultaneously predicting multiple outputs of blood sample frequencies with variable lengths using patient-specific features at the time of admission to the ICU.

PART I LITERATURE REVIEW

CHAPTER 2 ICU PATIENT MONITORING

In this section, we will provide background information on patient monitoring in the ICU, including the types of data collected, challenges faced, and traditional approaches to reduce and monitor phlebotomy in the ICU. We will also explore the advancements in the integration of predictive models into health care to provide the context for understanding the current state of the field.

2.1 Introduction to Patient Monitoring in the ICU

In an ICU, patients are connected to a range of devices, tubes, and monitors. These systems continuously provide critical data on the patient's physiological functions and guide their care. Some equipment may temporarily take over essential functions like breathing, nutrition, and urination until the patient can resume these tasks independently. For example, a patient with a traumatic brain injury (TBI) receiving care in the Neuro Science Intensive Care Unit (NSICU), a specialized ICU division, relies on this extensive monitoring, as shown in Figure 2-1. for identifying signs of deterioration, preventing complications, and enhancing outcomes. However, despite these benefits, the monitoring practices, including frequent blood sampling, present challenges that affect patient comfort and clinical results. These issues will be thoroughly explored in the following subsections.

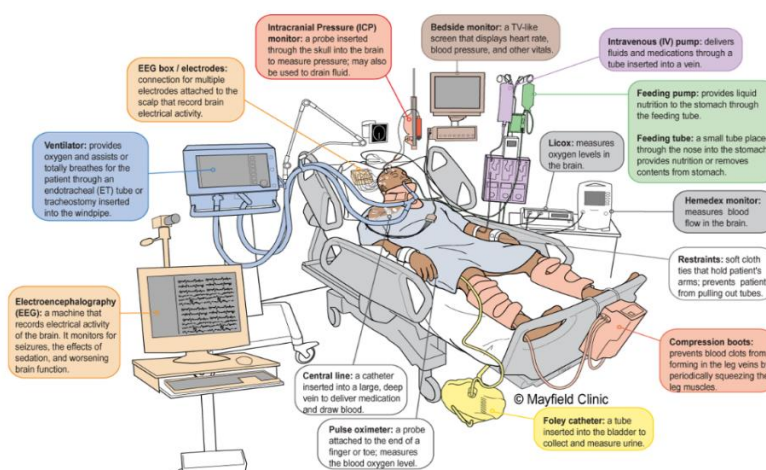


Figure 2-1: Multiple monitoring devices connected to a patient who suffered TBI¹

¹ Mayfield Brain and Spine Clinic, <http://www.mayfieldclinic.com/PE-TBI.htm>

2.1.1 Role of Blood Sampling in the Management of ICU Patients

Blood sampling is a fundamental practice for monitoring patients in the ICU. It plays a crucial role in healthcare decision-making, with laboratory data from these samples influencing approximately 60% to 70% of clinical decisions [1]. The results of these tests are essential for diagnosing underlying conditions, identifying infections, detecting organ dysfunction, and assessing metabolic abnormalities. Continuous monitoring of blood parameters allows real-time tracking of a patient's condition, enabling healthcare providers to assess treatment efficacy, detect complications early, and make timely interventions to improve patient outcomes [4], [15], [16].

To give an example, blood tests offer crucial insights into the function of vital organs like the liver, kidneys, heart, and lungs by detecting abnormalities in markers such as liver enzymes, creatinine, and arterial blood gases. They also monitor electrolyte balance and hematologic status, providing information about conditions like anemia or clotting disorders. Blood tests are also vital in evaluating the need for transfusions and provide prognostic information by tracking trends in lab values over time, guiding patient care decisions throughout their ICU stay. They also allow for targeted antibiotic or antifungal therapy, which is more effective than broad-spectrum treatments. In other words, as stated by the authors in [1], *“healthcare systems cannot operate without the information clinical laboratories provide.”*

However, the challenge remains in finding the right balance between frequent phlebotomy and obtaining necessary clinical results. In many ICU settings, blood sampling is primarily conducted either as part of routine care or at the request of a medical doctor [4]. Therefore, research has concentrated on the need to minimize unnecessary blood draws. Reducing these draws can significantly enhance patient comfort, conserve critical resources, and decrease the risk of iatrogenic anemia. Thus, a well-monitored approach to blood sampling is essential for ensuring that tests are conducted only when necessary, which would enhance patient outcomes, optimize the use of hospital resources, and streamline overall ICU operations related to blood testing.

2.1.2 Modern blood sampling techniques and practices

Raul *et al.* [2] established *“that there was a statistical significance in the production of iatrogenic anemia due to the daily sampling for laboratory monitoring and critical procedures in intensive*

care units” and that “*the implementation of patient blood management programs could address these issues*”. As highlighted, blood sampling is essential for monitoring and diagnosing ICU patients, but it must be managed carefully to prevent adverse effects from excessive use. In modern hospitals, healthcare providers follow strict protocols to reduce the frequency and volume of blood sampling focusing on patient comfort using efficient practices and advanced technology. Techniques such as using arterial lines, point-of-care testing, and micro-sampling are employed to reduce the need for repeated blood draws [3, 6, 7]. Table 2-1 highlights the techniques used to reduce blood sampling. Faisal *et al.* [17] implemented an 8-week quality initiative aimed at reducing unnecessary blood tests in three internal medicine services which included educational sessions, reminders, and posters, resulting in a 50% reduction in the median number of blood tests per patient per day and a 21% decrease in median hospital stay length. The intervention demonstrated that targeted education and cost-awareness can significantly decrease routine blood testing, though ongoing reminders are crucial for maintaining these improvements. In another study [5], the authors assessed the daily phlebotomy volume and associated blood loss in 400 patients undergoing major cardiac or vascular surgery. It found significant blood loss from diagnostic tests, with mean losses ranging from 167.9 mL to 695.3 mL for cardiac surgery patients and 80.5 mL to 470.3 mL for vascular surgery patients, depending on the length of hospital stay. The use of small-volume tubes and closed blood collection devices was shown to significantly reduce blood loss, with reductions of up to 824.0 mL for longer ICU stays. This highlights the potential of these interventions to mitigate blood loss and prevent waste in hospitalized patients. Similar findings were addressed by the authors in [3]. The analysis found moderate evidence that devices returning blood from venous or arterial lines can reduce blood loss by about 25%, and that bundled interventions including these devices also showed promise. These blood conservation strategies such as small volume tubes, closed blood sampling devices, decision tools for blood draw amounts, and non-invasive testing methods, among others prove to play a pivotal role in blood sampling reduction and the overall blood management system in ICU. The authors in [7] reviewed 39 trials across neonatal, pediatric, and adult populations to assess interventions for reducing diagnostic blood loss, revealing that evidence is highly variable. Blood conservation devices and smaller blood tubes significantly reduced blood loss and transfusions in adults, while point-of-care testing was effective in both adults and neonates. In summary, implementing bundled interventions

including staff education and protocols is essential for reducing the frequency and volume of blood tests in both adults and children.

However, in a recent cohort study of critically ill adults [6], it was found that patients experienced a median of 41 laboratory draws totaling 232 milliliters, with waste accounting for 10.8% of this volume even with bundles of techniques. Surgical patients and those in certain ICU settings had higher phlebotomy volumes. More intensive phlebotomy practices were linked to lower hemoglobin levels, longer hospitalizations, and increased red blood cell transfusions, with annual costs exceeding \$15 million. Phlebotomy remains a significant source of blood loss in critically ill patients, highlighting the need for improved practices and monitoring techniques. Furthermore, human error can also occur when information gets lost during handovers between shifts, leading to repeated phlebotomy, or when less experienced clinicians make decisions differently from more seasoned professionals. Some doctors may also conduct tests based on immediate judgment rather than adhering strictly to guidelines, despite being aware of the risks of frequent phlebotomy. While significant progress has been made in reducing unnecessary blood sampling, there remains room for improvement in how healthcare teams manage blood draws in critically ill patients. Implementing personalized predictions for blood sampling would complement existing interventions and further optimize patient care.

Table 2-1: Comparison of various strategies for reducing blood loss during blood sampling in ICU

Method	Description	Evidence of Effectiveness
Small-Volume Tubes	Tubes designed to collect smaller volumes of blood compared to standard tubes, reducing overall blood loss.	Studies show that using small-volume tubes can significantly reduce the total volume of blood drawn, with reductions in blood loss of up to 824.0 mL in longer ICU stays [5]
Closed Blood Collection Devices	Devices that allow for the collection of blood samples without exposing the blood to the environment, minimizing waste	Evidence indicates that closed systems can reduce blood loss by about 25% and lower iatrogenic anemia risk [3], [5].

	and reducing the need for multiple punctures.	They also help prevent contamination and improve sample integrity.
Decision Tools for Blood Draw Amounts	Algorithms and decision support tools that help clinicians determine the minimum necessary volume of blood to be drawn, based on the required tests.	Implementation of decision tools has shown to decrease the number of unnecessary blood draws and reduce total blood loss, contributing to improved patient comfort and resource use [7].
Arterial Lines for Continuous Monitoring	Inserting a catheter into an artery for continuous measurement of blood gases and other parameters, reducing the need for frequent blood draws.	Arterial lines reduce the frequency of blood sampling by providing ongoing monitoring, which is especially beneficial in critically ill patients [6]
Point-of-Care Testing	Testing methods performed at the bedside using portable devices that provide rapid results, minimizing the need for traditional lab-based tests.	Point-of-care testing reduces the frequency of blood draws needed for monitoring, which helps in managing patient comfort and efficiency in ICU settings [10].
Micro-Sampling Techniques	Techniques that use devices to collect very small amounts of blood, such as fingerstick methods or specialized micro-sampling devices.	Micro-sampling has been shown to significantly reduce the volume of blood drawn while still providing sufficient information for diagnostic purposes [7].

2.1.3 Influence of AI in ICU Patient Management

Artificial intelligence (AI) has profoundly transformed numerous aspects of today's life. Its impact is evident across various domains, including finance, economy, psychology, transportation, digital platforms, real estate, commerce, and healthcare, among others [18]. A study published in the Journal of Computational Science highlights AI's transformative impact, particularly in enhancing diagnostic accuracy by precisely distinguishing between viruses and bacteria [19]. In emergency

medicine [20], AI supports various clinical applications, including triage and disposition, disease-specific risk stratification, medical imaging, and emergency department operations. The potential benefits of ML in predictive medicine include early detection of patient deterioration and more personalized treatment.

Likewise, in ICU settings, studies focused on developing AI algorithms for mortality prediction, sepsis detection, length of stay estimation, early prediction of circulatory failure, and clinical decision support. In fact, in a review published by *Jiayi et al.* [21], AI models have shown significant promise in predicting sepsis mortality by analyzing patterns in electronic health records and vital signs. In another research, AI systems were used for sepsis detection and utilized ML algorithms to identify early signs of sepsis from a vast array of data, including lab results and vital signs, often with greater accuracy and speed than traditional methods [22]. Early detection is crucial for initiating prompt treatment, which can be lifesaving. AI also helps estimate patient length of stay in the ICU by analyzing historical data and patient characteristics, aiding in resource planning, and improving patient flow [23]. In this study by the authors in [24], the random-forest algorithm achieved impressive performance metrics - 98.04% accuracy, 99.8% precision, 99.9% sensitivity, and more, realizing a 2% improvement over existing models in predicting ICU length of stay and mortality. Furthermore, early detection of circulatory failure was performed by AI models [25]. Researchers developed an ML-based early-warning system that effectively predicts 90% of circulatory failure events in intensive care patients, with 82% of predictions made over 2 hours in advance. This system, which integrates data from multiple organ systems, significantly reduces false alarms compared to traditional methods and was validated with an independent patient cohort. These predictive tools have the potential to alert medical staff to high-risk patients, enabling timely interventions and potentially improving outcomes. These examples highlight how AI advancements can significantly improve critical care by offering insights that lead to better clinical decisions, optimizing patient management, and enhancing monitoring capabilities. Since patient monitoring is crucial throughout all phases of care, such innovations play a key role in advancing predictive medicine, allowing for more accurate and proactive interventions in ICU.

Given these advancements, addressing the issue of frequent blood draws in ICU settings is crucial, as they can cause patient discomfort, increase the risk of iatrogenic anemia, and lead to unnecessary use of resources. Based on patient data and trends, AI, or more specifically ML, can help predict when blood samples are needed, thus reducing their frequency while maintaining accurate and timely monitoring. We can lessen patient discomfort, minimize strain on healthcare resources, and enhance overall patient care. Improvement in blood test management could well be integrated into Clinical Decision Support Systems (CDSS) for monitoring phlebotomy in ICUs.

Recent efforts to reduce blood sampling have involved AI. However, the literature features only a few studies in this area, all focusing on developing predictive models that forecast future test results based on previous results in a time series. In 2011, F. Cismondi *et al.* [26] have presented their work at the IEEE International Conference on Fuzzy Systems, marking the first instance of applying this approach. At first, their focus was on predicting laboratory tests results. Using data from 26,665 patient records at Beth Israel Deaconess Medical Center, including a subset with gastrointestinal bleeding, the researchers employed various soft computing and statistical tools to analyze test usefulness. They found that urine output was the best predictor for the accurate results of hematocrit tests. Their models, which included fuzzy models and neural networks, accurately classified the usefulness of these tests up to 81% of the time. Two years later, the same authors conducted a study [11] to reduce unnecessary laboratory testing in ICU patients always in gastrointestinal bleeding patient by predicting which future tests would be most valuable for clinical decision. Utilizing data from 746 patients, they applied AI, again using fuzzy modeling, to assess 11 input variables, such as vital signs and previous lab results, to evaluate the utility of eight different lab tests. They achieved over 80% accuracy in classifying necessary versus unnecessary tests and reduced lab testing by 50%. The study highlights the effectiveness of AI in optimizing lab test use and its potential for broader application in different medical conditions using patient specific medical information. While both summaries focus on reducing unnecessary lab testing using AI, the previous summary emphasized the overall accuracy of predicting test usefulness (up to 81%) with a focus on hematocrit tests specifically. This summary, however, details the use of fuzzy modeling for predicting the value of various lab tests in a larger ICU dataset, achieving higher accuracy and a greater reduction in testing frequency, and outlines future exploration for broader

applications. This marked the beginning of a novel research area, and it paved the way for further predictive models to reduce unnecessary phlebotomy.

With the increasing availability of extensive patient data from electronic health records and improved documentation of vital signs and laboratory tests, researchers increasingly focused on minimizing blood sampling. Yu *et al.* published two articles on using Long-Short-Term-Memory (LSTM) solutions to recommend laboratory reduction strategies. In [13], their paper introduced a novel deep learning approach with a compact architecture to predict which future lab tests can be skipped and estimate their values based on observed data. The method successfully omitted 15% of lab tests with only a 5% loss in prediction accuracy. While focused on lab test results, this framework is broadly applicable to various decision-making problems. Continuing along this trajectory, in their second paper [11] they tried to improve the performance of their model by adding more data points in an attempt to deploy for generalization always utilizing spatial-temporal correlations. The model, validated on a database with over 4.5 million observations, treats lab testing as a series of decisions considering context over time and across modalities. It achieved a 20.26% reduction in laboratory tests while maintaining high accuracy (98.27%) in predicting test normality/abnormality and effectively recommending transitions. The model outperformed baseline approaches, demonstrating robust and iterative prediction capabilities for optimizing lab test policies. In contrast to their previous results, this paper highlights a global prediction model that integrates spatial-temporal correlations, showcasing a broader and more contextually aware approach to lab test improvement.

Another study [27], although it focused on redundant laboratory blood tests, improved predictions of abnormal test results by incorporating conditional entropy and conditional probability into machine learning models as additional features. Using an ICU dataset from Alberta, Canada, with 55,689 admissions, two approaches were compared: one using standard patient features and another including the new conditional features. The enhanced approach, which incorporated these new features, achieved a higher average Area-Under-the-Curve (AUC) of 0.89 compared to 0.86 with the standard approach. The pretest probability feature was particularly significant in enhancing model performance. In summary, while both approaches focus on optimizing laboratory testing in ICUs, this one emphasizes enhancing prediction accuracy through specific statistical features,

whereas the previous study highlighted a deep learning approach for overall lab test reduction. Other articles also focussed on implementing ML models “*to be able to quantify the level of uncertainty and expected information gained from diagnostic tests explicitly encourage useful testing and discourage low-value testing*”, [28], [29], [14]. Notably, in [28], the authors chose not to include age and gender in their analysis, although they recognized these factors could influence prediction outcomes. Their findings suggest that variables like age, sex, and medications taken by a patient could impact results. Therefore, predefined boundaries could limit the performance of the predictive system, and our research aims to address these limitations. Additionally, in [10], the authors noted that magnesium is often ordered independently, making it a candidate for exclusion in their analysis of predicting blood test results.

Essentially, AI in blood sampling management has primarily focused on developing tools for temporal prognostics, which advise doctors on whether a particular blood test should be omitted at a specific time based on temporal data from previous tests or patient conditions. However, the focus should shift towards creating models that predict the optimal frequency and sequence of blood sampling upfront at ICU admission. This innovative approach transforms blood test management in the ICU and enhances patient care, ultimately revolutionizing clinical decision-making in critical environments and serving as a valuable complement to existing methods.

2.2 Medical Patient Databases

In this sub-section, we will examine different types of medical patient databases, including open-source options, and discuss their advantages for research fields like machine learning and predictive medicine. This review aims to identify a suitable existing database from which we can extract specific features and label the data for the task focused on predicting blood sample frequencies. Given that machine learning relies heavily on quality data [30], finding an appropriate database is crucial for the success of our predictive model.

2.2.1 Review of Key Database Types and Their Applications

Medical patient databases are structured repositories of patient information collected from various sources. These databases play a crucial role in healthcare systems practice by providing comprehensive data that supports a wide range of activities. They are essential for patient care as

they enable clinicians to access detailed patient histories, track treatment outcomes, and personalize medical interventions. In research, these databases facilitate the analysis of large datasets to uncover trends, identify risk factors, and develop new treatments. Predictive modeling benefits significantly from these databases, as they allow for the creation of algorithms that can forecast patient outcomes, optimize resource allocation, and improve preventive care strategies. Medical patient databases come in several key types, each serving a unique role in healthcare research and practice. In a review paper by Yang *et al.* [31], the authors introduced a variety of publicly available databases for research purpose. Given the challenges of finding databases due to ethical considerations, this review helps us identify the different types of medical patient data available and assess their relevance to our task.

First, Electronic Health Records (EHRs) are among the most widely used and essential databases. EHR systems compile comprehensive patient information from various healthcare interactions, including medical history, treatment plans, and test results. They provide a holistic view of a patient's health over time, facilitating better coordination among healthcare providers and supporting informed clinical decision-making. By enabling real-time access to a patient's medical information, EHRs enhance diagnostic accuracy, improve treatment outcomes, and streamline administrative processes, making them indispensable in modern healthcare systems like hospitals. For our specific task, EHRs contain the exact information we need: detailed records of patient encounters, including the number of blood draws, the results of those draws, and the timing of each procedure. This data is crucial for our analysis.

Other types of medical databases, such as Clinical Trial Databases and Public Health Databases, play significant roles in healthcare research. Clinical Trial Databases offer a centralized platform for tracking and sharing information about clinical trials, containing detailed records of ongoing and completed studies, including trial design, objectives, methodologies, and outcomes. They are essential for researchers looking to review and build on existing studies, for healthcare professionals to stay informed about new treatments, and for patients seeking opportunities to participate in clinical research. By cataloging and making this data accessible, clinical trial database promote transparency, support evidence-based medicine, and expedite the development of new therapeutic options. Similarly, Public Health Databases, maintained by organizations such as the

National Health and Nutrition Examination Survey (NHANES), are crucial for monitoring and analyzing public health trends. They compile extensive data from surveys and health assessments, providing valuable insights into health status, behaviors, and environmental factors affecting populations. These databases support public health research by identifying emerging health issues, guiding policy development, and evaluating the effectiveness of public health interventions. By offering a broad view of population health, public health databases play a key role in shaping strategies for disease prevention, health promotion, and resource allocation. However, neither of these database types provides the specific information we are seeking for our task.

The authors in [31] introduce the Medical Information Mart for Intensive Care (MIMIC), which is an open-source database and includes a large amount of de-identified health data from critically ill patients admitted to the ICU at the Beth Israel Deaconess Medical Center. MIMIC contains a wide range of data, including vital signs, laboratory results, medications, and clinical notes, which are valuable for research and analysis in critical care. Although it includes similar types of information found in EHRs, its primary function is to act as a robust research resource rather than a clinical tool used in real-time patient care. Given its comprehensive nature and focus on ICU patients, MIMIC is an excellent candidate database for our study. The authors emphasize that MIMIC database provides an excellent support for all aspects of clinical research.

2.2.2 Exploiting MIMIC for Progress in Critical Care Research

The Medical Information Mart for Intensive Care MIMIC [32] is a comprehensive, openly accessible database that contains de-identified health data from patients admitted to the critical care units of the Beth Israel Deaconess Medical Center. Over the years, multiple versions of MIMIC have been released. The latest version, MIMIC-IV includes detailed records of vital signs, such as heart rate, blood pressure, and respiratory rate, which are continuously monitored throughout a patient's stay in the ICU. Additionally, MIMIC-IV includes data on laboratory results, which offer insights into blood test outcomes at various intervals, including the timing of blood test results. This depth of information enables researchers to analyze patient conditions and treatment responses with high precision. The presence of detailed timestamps and a diverse dataset from various ICU settings improves our understanding of the intricate interactions and outcomes in critical care, supporting the credibility and reliability of the current research.

In fact, MIMIC-IV is instrumental in a variety of research applications within ICU settings. Researchers can utilize the detailed timestamps and results of blood tests to study patterns and trends in patient health, monitor the effects of interventions, and predict potential complications. For example, the precise timing of blood samples combined with vital sign data allows for the development of predictive models that can anticipate patient deterioration or response to treatments. This capability supports the evaluation of clinical practices, improvement of treatment protocols, and overall enhancement of patient care. By leveraging the rich dataset provided by MIMIC-IV, we can drive evidence-based improvements in ICU management and refine predictive algorithms to better support clinical decision-making.

MIMIC-IV is accessible through the PhysioNet platform, which ensures that data use adheres to rigorous ethical and privacy standards. Researchers must complete a mandatory data privacy training course and sign a data use agreement before gaining access. This process guarantees that the data is used responsibly while maintaining patient confidentiality. Once access is obtained, researchers can perform a range of analyses, including statistical evaluations, ML model development, and outcome assessments. The structured format and detailed nature of MIMIC-IV² make it a valuable resource for advancing intensive care research and improving patient outcomes through informed decision-making and enhanced clinical practices.

2.3 Understanding Multi-Output Regression

Supervised models in the realm of machine learning can be categorized into four types based on the nature of their inputs and outputs. “*One-to-one prediction*” involves straightforward regression or classification tasks where a single input corresponds to a single output. “*One-to-many prediction*” occurs when a single input generates a sequence of outputs, such as in image captioning, where one image produces a sequence of descriptive words. “*Many-to-one prediction*” involves multiple inputs leading to a single output, as seen in sentiment analysis where multiple pieces of text are analyzed to produce a single sentiment score. “*Many-to-many prediction*” is the most complex, involving both multi-input and multi-output data, such as in machine translation,

² Medical Information Mart for Intensive Care, Freely available medical data for research, [MIMIC \(mit.edu\)](https://mimic.mit.edu/)

where a sequence of words in one language is converted into a sequence of words in another language.

As a preliminary analysis, we defined predicting blood sampling frequencies as a “*many-to-many prediction*” problem. Specifically, it involves a single input, which includes many static features (e.g., gender, age, vital signs, primary diagnoses, and previous blood test results), and applies multivariate regression to generate a sequence of dependent outputs such as the timings of blood draws throughout the ICU stay (t_1 to $t_{discharge}$ where $t_{discharge}$ represents the timing of the last blood draw before discharge), with t_0 representing the initial timestamp and serving as the input. Given this complexity, it is essential to employ ML models that can effectively handle sequential dependencies and manage the variability in output lengths. In all case, the aim is to identify a function that can accurately predict a timeline of blood sampling events based on a training dataset D of N samples. Each sample consists of static features $x \in \mathbb{R}^p$ and their corresponding timestamp vector $t \in \mathbb{R}^{discharge}$. Therefore, the dataset is represented as $D = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$. Each instance is characterized by an input vector of p features $x^{(1)} = (x_1^{(1)}, \dots, x_i^{(1)}, \dots, x_p^{(1)})$ and an output vector of variable *discharge* target outputs $t^{(1)} = (t_1^{(1)}, \dots, t_j^{(1)}, \dots, t_{discharge}^{(1)})$, with $i \in \{1, \dots, p\}$, and $j \in \{1, \dots, discharge\}$, and $(l) \in \{1, \dots, N\}$. This function should consider all the features in x and produce a prediction for each target variable in t :

$$h : \Omega x_1 \times \dots \times \Omega x_p \rightarrow \Omega t_1 \times \dots \times \Omega t_{discharge} \quad (4.1)$$

$$x = (x_1, \dots, x_p) \rightarrow t = (t_1, \dots, t_{discharge}), \quad (4.2)$$

where Ωx_i and Ωt_j denote the sample spaces of each predictive variable x_i , for all $i \in \{1, \dots, p\}$, and each target variable t_i , for all $j \in \{1, \dots, discharge\}$, respectively. Note that, all target variables are continuous meaning the timing of blood draws or samples could be any real number within the ICU stay period. The learned multioutput model will be used afterward to simultaneously predict the values of all target variables $\{\hat{t}^{(N+1)}, \dots, \hat{t}^{(N+1)}\}$ of the new incoming unlabeled instances $\{x^{(N+1)}, \dots, x^{(N+1)}\}$. [33]

Given that the output targets can be considered as sequential time series data, LSTM networks are especially well-suited for this task [34]. Even if the current task involves short-term dependencies

at the time of admission to the ICU, testing LSTMs can help assess whether the advanced architecture can capture subtle temporal patterns better than simpler models. If the task or dataset changes in the future to include longer sequences or more complex temporal dependencies, LSTMs could be better suited for handling such scenarios. Testing them now prepares for potential future expansions and scalability. Even with one timestamp of input, LSTMs can still leverage their gating mechanisms to better understand the sequential nature of the data compared to simpler models and might offer better robustness to noise and variability in the data. This can be beneficial in real-world scenarios where data quality might be inconsistent [35] but can be adapted for multi-target regression by treating different targets as separate output dimensions. By utilizing a single timestamp as input, this approach serves as a strategic entry point for exploring complex predictive tasks.

The following sections discuss various ML approaches and models that are suited for “*many-to-many*” predictions, including LSTMs. We will explore their strengths, limitations, and applications in predicting phlebotomy sequence in ICU settings, setting the stage for the development and testing of an LSTM model tailored to this specific task.

Given the complexity of predicting blood sampling frequencies with variable output lengths and sequential dependencies, exploring a range of modeling approaches is crucial. Multi-output regression builds upon existing single-output models by adapting them to handle multiple outputs simultaneously. Historically, single-output regression models have evolved through various methodologies, each addressing specific challenges in predictive accuracy and complexity. This foundation paved the way for multi-output regression, which adapts these established approaches to simultaneously predict multiple outcomes. The timeline presented in Figure 2-2 showcases the evolution from early foundational models and traditional machine-learning techniques to modern, sophisticated models. Each era brought innovations that addressed the limitations of previous approaches, leading to the advanced machine-learning techniques we use today, including both classical methods and deep-learning models.

In the 1950s and 1960s, linear regression laid the groundwork for regression tasks, with multivariate linear regression extending these models to handle multiple dependent variables.

While straightforward and computationally efficient, these methods are limited by their inability to model complex, non-linear relationships and tend to exhibit high bias. They remain a good baseline model for analysis and comparison. The 1970s and 1980s introduced polynomial regression, which could address non-linear relationships but often struggled with overfitting and high computational costs. The introduction of decision trees, in the 1990s, was capable of handling multi-output regression by splitting data based on feature values, allowing it to predict multiple outcomes simultaneously by following different branches for each output. While decision trees are interpretable and flexible, they can become overly complex and prone to overfitting. This is why random forests enhanced generalization by introducing uncertainty. The 2000s saw the adaptation of Support Vector Machines (SVMs) for multi-output tasks, offering strong performance for regression problems but requiring careful tuning and handling of large datasets. Ensemble methods like Gradient Boosting Machines (GBM) and AdaBoost further refined multi-output regression by combining predictions from multiple models, but they can be sensitive to noisy data. The 2010s marked the rise of deep learning models, including Deep Neural Networks (DNNs), which can handle complex multi-output regression through multiple output layers. These models are powerful but often require large datasets and extensive training. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks excelled in modeling sequential data and long-term dependencies. The introduction of Gated Recurrent Units (GRUs) provided a simpler alternative to LSTMs, with improved efficiency but similar limitations in handling long sequences. The introduction of the Transformer model in 2017 modernized sequence processing by eliminating the need for recurrence and relying entirely on self-attention mechanisms, vastly improving the efficiency and scalability of multi-output tasks. Models like T5 (Text-To-Text Transfer Transformer) and BART (Bidirectional and Auto-Regressive Transformers) unified various natural language processing tasks into a text-to-text framework, enhancing the versatility and performance of sequence-to-sequence models but still face challenges related to scalability and efficiency and increased computational demands. Finally, OpenAI's series of Generative Pre-trained Transformers (GPT) further demonstrated the power of these architectures in generating coherent and contextually relevant text, with significant milestones such as GPT-1 in 2018, GPT-2 in 2019, GPT-3 in 2020, and GPT-4 in 2023. Each model has uniquely contributed to the field of multi-

output regression, with ongoing research focused on optimizing their strengths while addressing their limitations.

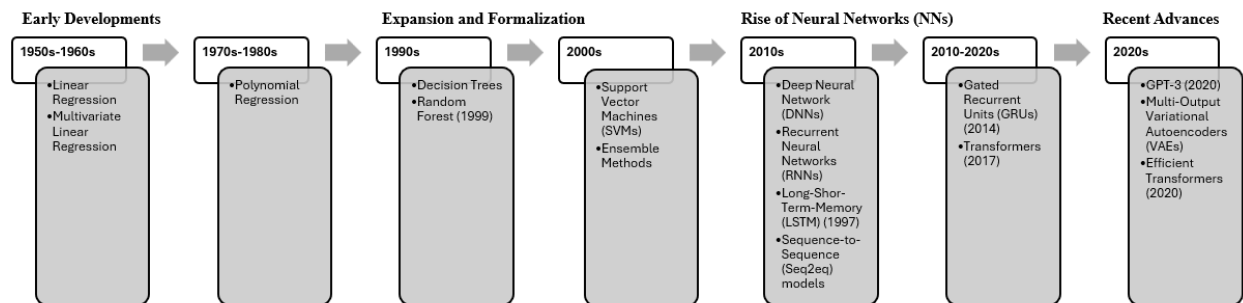


Figure 2-2: Timeline of Multi-Output Regression Models

Following this historical timeline, multi-output regression can be approached in several ways, supported by a vast body of literature [33], [36]. Thus, we can categorize multi-output regression into three main groups: models that naturally support multi-output regression, multioutput regressor, and regression chains. Within these categories, we find various approaches and techniques tailored to specific use cases. For instance, models that naturally support multi-output regression, like random forest and ensemble methods, can directly predict multiple outputs simultaneously. In contrast, multi-output regressors that extend single-output models often involve adaptations of linear regression or support vector machines, enabling them to handle multiple targets by leveraging shared information. Lastly, regression chains provide a sequential framework where the prediction of one output can influence the next, effectively capturing interdependencies among the outputs. While some techniques, detailed in Table 2-2, are more naturally aligned with certain groups, many can be adapted for use across multiple categories in multi-output regression. Thus, the choice of technique is heavily influenced by the specific task and the nature of the problem being addressed. Our way of grouping these multi-output regressions is represented in Figure 2-3.

Table 2-2: Overview of multioutput regression techniques and their applications

Technique	Description	Classical Models	Neural Networks
-----------	-------------	------------------	-----------------

Model Stacking	Train separate models for each output and combine their predictions. This can help capture relationships between outputs.	Linear Regression, Decision Trees	Neural Networks
Multi-task Learning	Use a single model that shares parameters across multiple outputs. This can improve generalization, especially when tasks are related [36]	SVR, Random Forests	Deep Neural Networks (DNNs)
Output Layer Modification	Change the output layer of a neural network to accommodate multiple outputs, ensuring the appropriate activation functions are used.	N/A	Neural Networks
Ensemble Methods	Combine predictions from multiple models trained on different aspects of the data. This can increase performance.	Random Forests, Gradient Boosting	N/A
Hierarchical Models	Structure predictions in a hierarchy, where some outputs depend on others, allowing for a more organized approach to multi-output learning.	Decision Trees	Graph Neural Networks

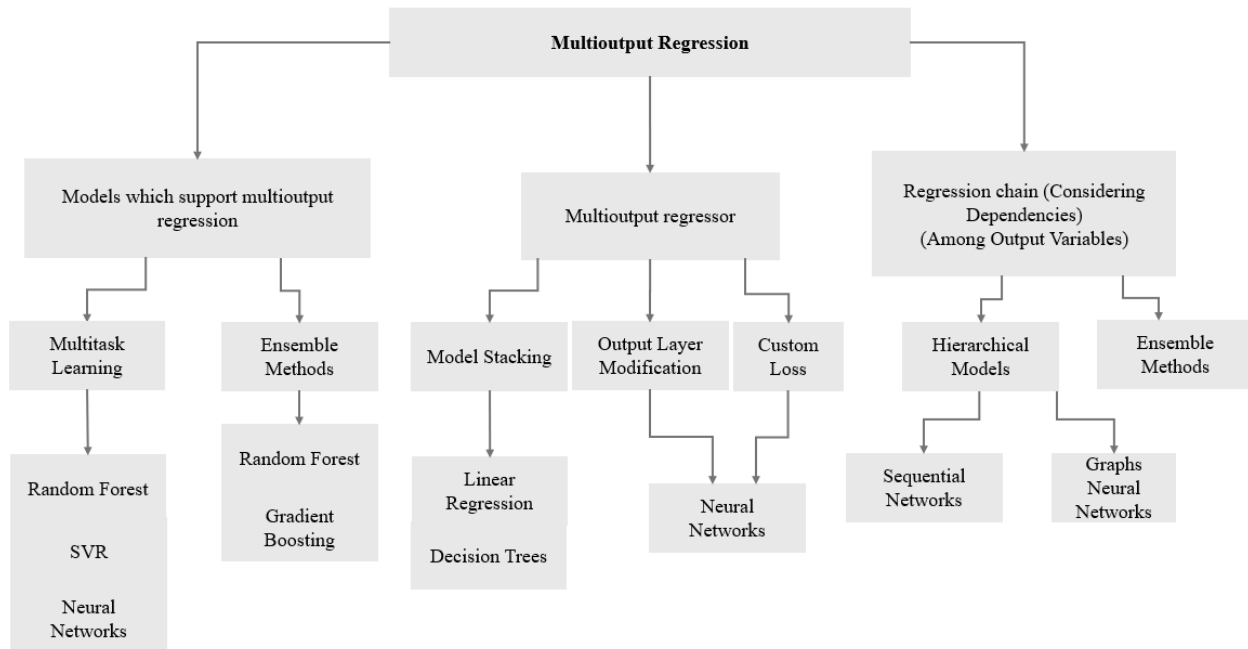


Figure 2-3 : Classification of multioutput regression models

2.4 LSTM Theory

Long Short-Term Memory (LSTM) models, a subset of neural networks, stand out and are well-suited for sequence-based tasks with time series data, especially given the challenges of predicting blood sampling frequencies with variable output lengths and sequential dependencies. Furthermore, the flexibility of LSTMs allows for extensive tuning of hyperparameters, such as the number of layers and units per layer. This adaptability enables them to be tailored specifically to our prediction task, enhancing performance over traditional regression models that may struggle with such complexities.

2.4.1 Introduction to LSTMs

LSTM networks are a special type of Recurrent Neural Networks (RNNs) designed to overcome their limitations, particularly the problem of vanishing and exploding gradients. Introduced by Hochreiter and Schmidhuber in 1997 [34], LSTMs are capable of learning long-term dependencies and maintaining context over extended sequences, making them particularly effective for tasks involving sequential data like predicting blood sample frequencies. In fact, they have internal

mechanisms called gates that can regulate the flow of information. These gates can learn which data of the sequence to keep and which data to throw away. By doing that, it learns to use relevant information to make predictions. Their effectiveness has been demonstrated across various applications, including speech recognition, text generation, video captioning, and time series analysis.

To put it simply, an LSTM can be compared to a storyteller who keeps a detailed notebook. As it processes each input, it decides what information to note down (output), what to erase (forget), and how to update its notes (update cell state). This allows it to remember important details over time and maintain coherence, even when dealing with lengthy narratives. Let's say now that our storyteller is also a doctor trying to figure out the best times to schedule a blood test for a patient based on their health data over time. This time, he has just been given a single detailed page from a patient's health. This includes key pieces of information about the patient's condition at one point in time. The LSTM, even with only one page, uses its internal memory to understand and remember the patterns and relationships within the page. This means it looks at the information and learns how similar past situations might lead to certain future events. Based on the patterns learned from many similar pages (from training data), the LSTM predicts multiple future timestamps for blood tests. It estimates when these future tests should occur by analyzing how the features on the current page relate to past patterns. For instance, consider a scenario where a dataset indicates elevated levels of a specific biomarker. An LSTM model, having been trained on numerous comparable samples, forecasts the necessity for blood tests at intervals of 3, 7, and 14 hours, thereby generating a tailored blood test schedule. LSTM's capacity to retain important details and recognize patterns from previous cases enables it to effectively convey the story of this new patient.

2.4.2 Understanding the Structure of LSTMs

LSTMs share the same control flow as neural networks (NNs). They all process data sequentially, passing information as it moves forward. The forward pass and backward pass, along with loss functions, input layers, output layers, and multiple hidden layers, remain the same. The key differences lie within the LSTM cells, which can be thought of as '*super neurons*' composed of multiple interacting components. The mathematical operations within each cell allow the LSTM to selectively retain or discard information.

As seen in Figure 2-4, the core concept of an LSTM is the cell state, which is influenced by gates (input, forget, and output) that manage the flow of information. These gates utilize sigmoid activation functions to assess the significance of the information. The outputs of these functions range from 0 to 1, guiding the decision to retain or discard data. Specifically, if the output is closer to 1, it indicates that the information should be kept, while an output closer to 0 suggests that it should be discarded. Each of the three gates has its sigmoid activation function. In addition to these, two other activation functions, typically tanh, are used to regulate the output and mitigate issues like exploding gradients during operations. Even with just one timestamp, the gating mechanism and cell state work to selectively manage information, allowing the LSTM to extract relevant information based on the input features.

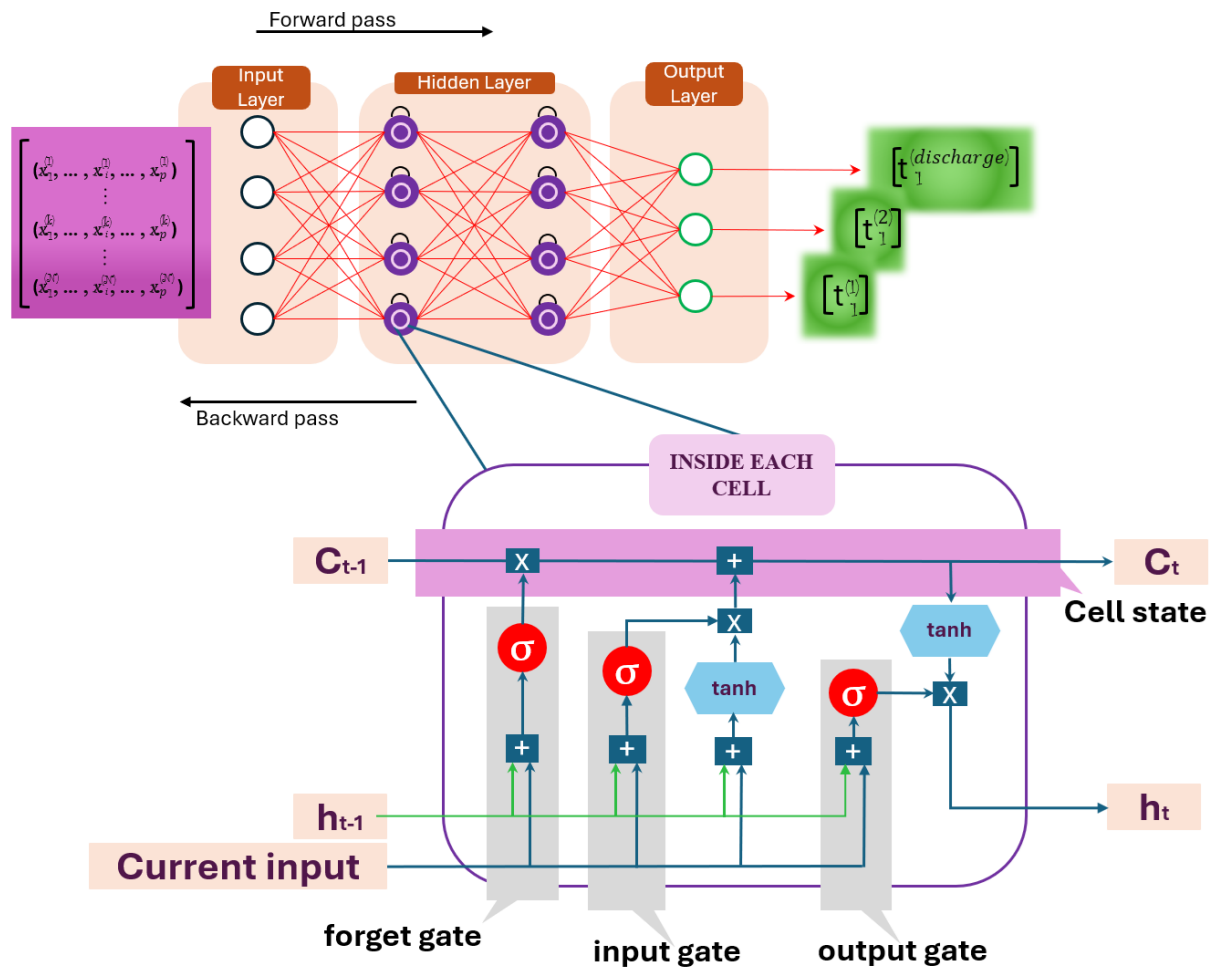


Figure 2-4: Complete LSTM architecture using patient input features $x_1, \dots, x_p$ and blood test frequencies $t_1, \dots, t_{discharge}$

The cell state (C_t) acts as a conveyor belt, running through the entire sequence with only minor linear interactions, thus preserving long-term dependencies. The hidden state (h_t) contains information about the sequence at each time step and is passed to the next module in the sequence. The input gate decides how much of the new information from the input to incorporate into the cell state. The forget gate decides what portion of the previous cell state to discard. The output gate determines what part of the cell state to output as the hidden state. This gate controls the flow of information from the cell state to the hidden state, which is used as the output at the current time step and passed to the next LSTM cell. The equations governing these gates and the updates to the cell and hidden states are as follows:

Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad 4.1$$

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad 4.2$$

Cell State Update:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad 4.3$$

New Cell State:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad 4.4$$

Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad 4.5$$

Hidden State:

$$h_t = o_t \odot \tanh(C_t) \quad 4.6$$

where the initial values are and and the operator denotes the Hadamard product (element-wise multiplication, defined for matrices of same dimensions) and the $\cdot$ refers to the usual matrix multiplication according to the rules of linear algebra.

While this complex structure may not operate at full capacity with only one timestamp as input, it can still learn from the data and perform the functions typical of a neural network. Testing it now will provide valuable insights into how LSTMs work for predicting blood test frequency in ICU. In this setup, the LSTM takes the single page of information (one timestamp with patient's features at admission time to the ICU) and uses its learned memory to predict multiple future points in time when blood tests might be needed $t = [[t_1^{(1)}], \dots, [t_1^{(j)}], \dots, [t_1^{(discharge)}]]$. This prediction is based on patterns it has learned from similar past situations, allowing it to provide forecasts even from limited input data. Furthermore, this layout could be extended to include the current literature of

predicting blood test results as well, integrating multiple timestamps as inputs once the predicted sampling frequency and its corresponding test results have been established.

During backpropagation in an LSTM with only one timestamp, the process resembles that of traditional neural networks, but there are specific considerations to keep in mind. With only one timestamp, the model receives a single input and computes one output, simplifying the sequence and eliminating any temporal dependence on previous time steps. However, the LSTM's architecture still allows it to learn from this isolated input. The backpropagation algorithm calculates gradients based on the loss between the predicted output and the actual output, and since there's only one timestamp, these gradients are computed solely from that instance. The gating mechanisms, input, forget, and output gates, will still be updated during backpropagation, with gradients flowing back through them to adjust their associated weights, even if the updates are based on limited information. Although the LSTM can learn patterns from this single input, it may struggle to generalize due to the absence of historical context provided by multiple timestamps, making it difficult to capture long-term dependencies. Consequently, the overall capacity of the LSTM to learn complex patterns may be underutilized, with the depth of learning not reaching its full potential. Nonetheless, this trade-off is one we are willing to accept to enable the prediction of variable-length blood test frequencies.

2.4.3 Impact of LSTMs on Multi-Output Regression Performance

Classic multi-output regression models typically rely on fixed architectures that struggle to adapt to the nuances of predicting variable-length output sequences, particularly in complex tasks like forecasting blood test frequencies. These models are often constrained by the requirement for fixed-size inputs, making it challenging to manage outputs that vary in length without extensive postprocessing. Furthermore, traditional regression approaches lack the flexibility to incorporate contextual information from previous outputs, limiting their capacity to capture the intricate relationships between input features and the timing of multiple output sequences. This limitation does not align with our objectives, as we aim to develop a model capable of accommodating more complex relationships. Specifically, we seek to extend its predictive capabilities to encompass existing literature on predicting test results as well. In contrast, neural networks, particularly

LSTMs, offer greater adaptability in handling variable-length outputs. They can effectively learn from a single input timestamp, enabling them to predict multiple future timings for blood tests by capturing the underlying patterns and complexities inherent in the data. This makes neural networks a more suitable choice for tasks requiring nuanced predictions, as they can leverage their structural flexibility to accommodate the variability in output sequences. In the literature, many techniques were employed to handle multi-output variable length sequence with neural networks. Table 2-3 presents a resume of the techniques that will be explored in our thesis to develop an LSTM model capable of prediction blood test frequencies.

Table 2-3: LSTM techniques for variable multi-output regression

LSTM Techniques	Description
Padding	Padding involves adding zeros or other values to sequences to ensure they all have the same length, which is crucial for batch processing in neural networks. In the context of predicting variable output timestamps from a fixed input timestamp, padding allows the model to handle different lengths of output sequences consistently.
End-of-Sequence (EOS) Tokens	Using special tokens to indicate the end of a sequence helps the model understand when to stop processing, particularly in variable-length outputs. In the context of predicting blood test frequencies, incorporating end-of-sequence tokens can improve the model's ability to recognize when to conclude its predictions for the output timestamps. During post-processing, an algorithm is employed to truncate the sequence upon detection of the EOS token.
Repeat Vector Layer	The RepeatVector layer is used to repeat the input sequence a specified number of times. This technique is useful in situations where the model needs to process the same input data multiple times, such as when generating output sequences that depend on a fixed input. For instance, in predicting blood test frequencies, this allows the model to utilize a single input timestamp to forecast multiple future outputs.

Addressing variable-length output sequences is a challenging task in machine learning, and it has been approached using a variety of techniques beyond those previously mentioned. The choice of method often depends on the specific task, the nature of the data, and the desired outcomes. As this research does not aim for replication, we will adapt these techniques to suit our specific objectives in predicting blood sample frequencies in the ICU. This flexibility makes LSTMs better suited for specific problems compared to traditional multi-output regression models. However, training LSTMs can be challenging due to the necessity of tuning multiple hyperparameters, including the number and types of layers, units per layer, learning rate, batch size, loss function and so on... Additionally, we cannot test all these methods simultaneously. Therefore, we must concentrate on one approach and implement improvements. Despite these complexities, we believe this focus is a valuable trade-off for establishing a robust baseline model for predicting blood test frequencies.

2.5 Evaluation Metrics and Loss functions

2.5.1 Key Metrics for Multi-Output Regression

Different metrics provide critical insights into how well the model generalizes to unseen data. Without robust evaluation metrics, it becomes challenging to gauge the true ability of a model, leading to poor outcomes in real-world scenarios. In a survey on multi-output learning published in the *IEEE Transactions on Neural Networks and Learning Systems*, Xu *et al.* [37] noted that the metrics commonly used for multi-output regression tasks are similar to those in single-output regression, including Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE, square root of MSE). For instance, MAE provides a more interpretable measure that may be preferred in clinical settings, while MSE highlights larger errors, which can be critical in situations where significant discrepancies could affect patient safety. Thoughtful consideration of these metrics is crucial, as a deep understanding is necessary for accurate interpretation in each specific context (see Table 2-4). In our case, it is particularly important to achieve high predictive performance not only for the timings of blood draws but also for the length of the sequences in other words, the total timestamps. We can apply these metrics to both dimensions and integrate their performance to provide a thorough evaluation of our predictive model.

Table 2-4: Evaluation metrics for multi-output regression

Metrics	Definition	Equation	Interpretation
Mean Absolute Error (MAE)	Average of the absolute differences between predicted and actual values.	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $	- Retains the same units as the original data, facilitating easy interpretation, - Treats all errors uniformly, making it less affected by outliers.
(Root) Mean Squared Error (MSE / (R)MSE)	Average of the squared differences between predicted and actual values.	$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$	- Highly sensitive to large errors, emphasizing them disproportionately. - Expressed in squared units; useful for optimizing and penalizing significant discrepancies
R-squared (R^2)	Proportion of variance in the dependent variable that is predictable from the independent variables.	$1 - \frac{\sum_1^n (y_i - \hat{y}_i)^2}{\sum_1^n (y_i - \bar{y}_i)^2}$	- Indicates the goodness of fit of the model - Ranges from 0 to 1, where 1 indicates that the model explains all variability in the response data around its mean. Higher values suggest better predictive power.

Understanding how a model reacts is just as important as its predictive performance. While metrics like MAE and RMSE provide insights into how well a model performs in terms of prediction accuracy, they may not fully capture the complexities of our task. We need to take into account both the averaging overall MAE and RMSE on each target values and the sequence length. We also need to consider a hybrid metric that can add another layer of interpretation. Therefore, we aim to define a new hybrid metric that incorporates aspects of both metrics while addressing their limitations, ultimately enhancing the evaluation framework for our multi-output regression task. The use of the combined metric will act as a guide for tuning the model. For instance, if the combined metric improves with adjustments to the model or data, it can be an indication of better overall performance. Again, for the specific task of predicting blood test frequency, it is essential to consider both the length of the sequence and the actual output values, which respectively is the total number of blood draw and the time at which blood draws should occur. This dual focus will ensure a more comprehensive evaluation of the model's performance, addressing both timings and accuracy in the predictions. As highlighted by the authors in [38] regarding evaluation of multi-

target regression, it is important to develop evaluation metrics that account for the relationships between multiple targets, rather than relying solely on aggregate performance metrics. They propose a more comprehensive evaluation framework that integrates various aspects of model performance, including predictive accuracy, interpretability, and the model's ability to generalize across targets.

2.5.2 Loss Functions and Their Influence on Inference

The choice of loss function can significantly affect model training dynamics and final performance. Always in the same survey, Xu *et al* [33] explained that “*various loss functions were defined to compute the difference between the ground truth and the predicted output. Different loss functions present different errors given the same data set, and they greatly affect the performance of the model*”. Traditional loss functions, such as Mean Squared Error (MSE) and Mean Absolute Error (MAE), are commonly employed in regression tasks. Like predictions, MSE is often favored during training for its sensitivity to larger errors, making it suitable for applications where outlier detection is crucial. However, its susceptibility to outliers can lead to skewed performance, particularly in medical datasets where extreme values may distort the learning process [39]. Furthermore, recent studies highlight the importance of testing different loss functions in neural networks to ascertain their impact on model performance. For instance, when predicting multi-output sequences, the choice of loss function can significantly influence the model's ability to generalize from training data to unseen instances. This adaptability becomes crucial when working with fixed input features while predicting variable output timings, as is the case in our study on blood sampling frequencies in ICU patients.

In contrast, Correntropy loss (CL) is particularly effective for scenarios where the data is influenced by non-Gaussian noise, as it can capture more complex relationships compared to traditional loss functions. In traffic flow prediction [35], where factors like irregular patterns or outliers can significantly impact the results, using CL loss function that accounts for these inner factors led to more robust predictions. Similarly, in medical data, noise can arise from various sources, such as measurement errors or variations in patient conditions. Fitting the loss function to minimize the impact of this noise allows the model to focus on critical patterns that are more indicative of the underlying health conditions. CL measures the similarity between the predicted and actual

distributions, allowing it to capture the underlying structure of data more effectively. By focusing on local similarities, CL enhances the model's ability to learn from limited or variable input sequences, making it well-suited for predicting blood test frequencies from a single timestamp input. It accounts for both mean and variance and captures local information around the predicted values. The formula is presented as follows:

$$C(y, \hat{y}) = - \sum_{i=1}^n K(y_i - \hat{y}_i) ,$$

$$K(x) = e^{\left(-\frac{x^2}{2\sigma^2}\right)},$$

where the parameter σ (sigma) controls the bandwidth of the kernel 4.7

By multiplying this expression by -1, the measure is transformed into a loss function, where lower values indicate better performance. Tuning the σ value is crucial, as it adjusts the sensitivity of the loss function where a smaller σ makes the model more responsive to minor discrepancies, while a larger σ can enhance robustness against noise. Overall, these adjustments aim to improve the effectiveness of CL in model training. Given the complexity of medical data and the nuances of our task, this literature underscores the necessity of experimenting with CL alongside traditional methods. By doing so, we aim to establish a strong baseline model that not only predicts blood test timings effectively but also demonstrates resilience to the irregularities often present in healthcare datasets.

2.6 Conclusion

LSTM networks represent a significant advancement in ML, particularly suited for predicting blood test frequency in the ICU. Their unique architecture enables them to retain long-term dependencies and effectively learn from sequential data, overcoming key limitations associated with traditional RNNs. LSTMs have shown versatility and efficacy across various applications, including natural language processing, time series prediction, speech recognition, and healthcare, solidifying their role as essential tools in the advancement of AI technologies. This research will focus on experimenting with LSTM networks by utilizing different evaluation metrics, exploring a hybrid approach, and testing various loss functions and will give insight into how LSTM models deal with this specific tasks. This study is the first to apply these methodologies for predicting

blood test frequency in the ICU at the time of admission. Focusing on a single timestamp is a strategic choice that allows us to begin our exploration while laying the groundwork for future expansions.

PART II CONTRIBUTION

CHAPTER 3 METHODOLOGY

The main objective of this research is to develop a predictive model for determining the frequency of blood sampling for ICU patients upfront. We can achieve this goal by experimenting with LSTM networks. This chapter provides an overview of the methodology employed to construct and optimize our ML pipeline for blood sample frequency prediction, Figure 3-1. In doing so, we will focus on the key goals outlined in the research objectives: identifying an existing dataset, selecting the most relevant features, defining the problem, evaluating the model's effectiveness, and investigating the impact of various loss functions.

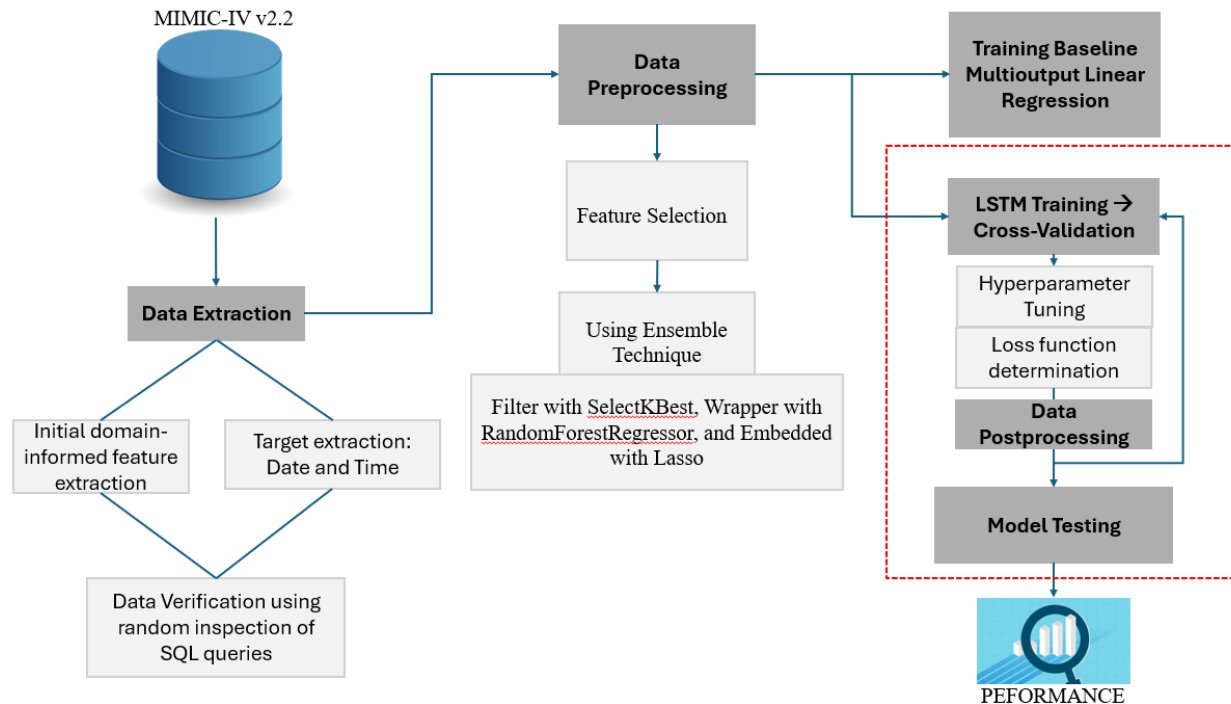


Figure 3-1: Overview of the methodology process for building a machine learning model for predicting blood test frequencies in the ICU

3.1 Dataset Creation

For supervised machine learning models, it is essential to create a labeled dataset. The initial step in this methodology involves identifying the relevant features and target values necessary for

accurate predictions. This section presents a comprehensive definition of the features and target values used in training the LSTM.

3.1.1 Feature Definition and Units

As discussed in earlier chapters, healthcare professionals use advanced monitoring devices and technologies to inform their decisions for stabilizing patients in the ICU. At the time of admission, they typically have access to essential information, including medical history, admission notes, vital signs, laboratory test results, imaging studies, and clinical assessments. These parameters are critical for initial evaluations and will serve as the foundation for our model's first implementation.

In selecting the features for this model, we drew inspiration from studies that identified 12 key laboratory test results, 7 vital signs, age, and gender as significant predictors for predicting blood test results [11], [13], [29], [40]. Although predicting blood test results differs from our tasks, these features are expected to be impactful in predicting the frequency of blood draws as well. Additionally, we included the primary diagnosis and the first care unit as they provide vital context for understanding the patient's condition at the time of admission. Other features such as insurance, marital status and race were also considered. Although we did not include imaging studies and certain clinical information, the selected features provide a solid foundation for identifying effective and ineffective elements of our predictive task. This selection is based on both empirical evidence and clinical insights, enhancing the model's potential relevance and accuracy. Table 3-1 details the features that will be used as input to our model. For a complete detail on the type and units of these features, refer to Appendix A - Table A. 1. We included a total of 5 features as demographics, 8 vital signs (adding weight as an indicator), 18 blood test results and 2 clinical assessment variables for a total of 33 input features. The target values consist solely of date and time, with varying lengths. These values are converted into intervals based on the timing of previous blood samples. The output represents a sequence of time intervals, where each interval indicates the difference between the current blood sample and the preceding one. For instance, if we have [0.2, 2, 1.2], this signifies that the first blood draw should occur 0.2 hours after admission, the second draw 2 hours after the first, and the third draw 1.2 hours after the second, always referencing the previous blood draw. In this scenario, the patient underwent three blood draws over

a total period of 3.4 hours. As an initial draft of the model, and for the sake of simplicity, this study focused on predicting the timing and frequency of blood tests rather than future blood test results. Consequently, the extraction of related blood draw values was not performed; instead, the emphasis was placed on the timing of each specific draw.

Table 3-1: Patient features and data identifiers in MIMIC-IV

Category	Features	Identifier - table in MIMIC-IV
Demographics	Gender (used in [11], [13])	gender - patients
	Age (used in [11], [13])	anchorage - patients
	Insurance	insurance - admissions
	Marital status	marital_status - admissions
	Race	race - admissions
Vital signs	Weight	patientweight - inpatientevents
	Heart Rate (used in [11], [13], [29], [40])	220045 - d_items
	Respiratory Rate (used in [11], [13], [29], [40])	220210 - d_items
	Oxygen Saturation pulse (SPO ₂) (used in [11], [13], [29], [40])	220277 - d_items
	Non invasive Mean systolic blood pressure (Bpsystolic) (used in [11], [13])	220179 - d_items
	Non invasive Mean diastolic blood pressure (Bpdiastolic) (used in [11], [13])	220180 - d_items
	Non invasive Mean blood pressure (Bpmean) (used in [11], [13], [29], [40])	220181 - d_items
	Temperature (based on TempSite) (used in [40])	223761 - d_items
Laboratory tests	Sodium (Na) (used in [11], [13])	50983 - d_labitems
	Potassium (K) (used in [11], [13])	50971 - d_labitems
	Chloride (Cl) (used in [11], [13])	50902 - d_labitems
	Bicarbonate – HCO ₃ (used in [11], [13])	50882 - d_labitems

	Calcium (Ca) (used in [11], [13], [29], [40])	50893 - d_labitems
	Magnesium (Mg) (used in [11], [13])	50960 - d_labitems
	Phosphate (PO4) (used in [11], [13])	50970 - d_labitems
	Creatinine (Cr) (used in [11], [13])	50912 - d_labitems
	Blood Urea Nitrogen (BUN) (used in [11], [13])	51006 - d_labitems
	Hematocrit (used in [29], [40])	51221 - d_labitems
	Platelet Count – Plt (used in [11], [13], [40])	51265 - d_labitems
	Hemoglobin – hgb (used in [11], [13], [40])	51222 - d_labitems
	White Blood Cells (used in [11], [13])	51301 - d_labitems
	Mean Corpuscular Hemoglobin Concentration (MCHC)	51249 - d_labitems
	Red Blood Cells (RBC)	51279 - d_labitems
	Mean Corpuscular Volume (MCV)	51250 - d_labitems
	Mean Corpuscular Hemoglobin (MCH)	51248 - d_labitems
	Red Cell Distribution Width (RDW)	51277 - d_labitems
Clinical Assessment	Primary Diagnosis	icd_code - diagnoses_icd (seq_num = 1)
	First Care Unit	first_careunit – icustays

This step is crucial for identifying the appropriate features for our model. Based on previous research, we have also opted to exclude magnesium from our study. We will explore this, along with the potential removal of other features, using feature selection techniques to enhance the model's performance and determine which features are beneficial or detrimental.

3.1.2 Source and Methods for Feature and Target

In this section, we outline where and how to obtain the features and target values necessary for our analysis. This thesis uses the fourth and latest version of MIMIC, MIMIC-IV v2.2, which incorporates numerous enhancements over its predecessors. MIMIC-IV was released in 2022 and includes data from patients admitted between 2012 and 2019, making it a recent database. One

significant improvement is the **icustays** table, which is derived from the **transfers** table and includes only patients admitted to the ICU. The study is limited to the context of adult ICU patients (> 18 years old). The database can be accessed through various methods, including download for use with PostgreSQL and other software applications. Additionally, PhysioNet offers access via Google Cloud's BigQuery, as well as through its own web-based interface, which allows for direct data exploration and analysis. After creating an account and meeting the ethical requirements, users can also utilize APIs for programmatic access and leverage the PhysioNet Archive for downloading datasets. For our queries, we specifically used Google Cloud's BigQuery platform for its efficiency and scalability.

The data extraction process primarily involves leveraging *SQL* queries. Additionally, *Excel* is helpful in organizing and establishing the variables essential for our analysis. This combined approach ensures a thorough data collection, aligning with the needs of our ML workflow and our research objectives. Significant effort was dedicated to obtaining a substantial amount of data, which will be further explored for potential data augmentation if necessary.

We use the following tables in the MIMIC-IV to extract the features and target values mentioned earlier for each patient admitted to the ICU:

- **patients**: Basic demographics and identifiers.
- **admissions**: Details of patient admissions.
- **diagnosis_icd**: ICD codes for diagnoses.
- **d_icd_diagnosis**: Descriptions of diagnoses.
- **chartevents**: Charted data from clinical observations.
- **d_items**: Definitions of recorded items.
- **procedureevents**: Details of procedures performed.
- **inpuvents**: Information on inputs like medications and fluids.
- **outputevents**: Data on outputs like fluid removal.

- **icustays**: Information on ICU stays.
- **labevents**: Laboratory test results.
- **d_lab_items**: Definitions of laboratory tests.

A relational schema is presented in Figure 3-2 to illustrate the connections between these tables and the queries derived from them. It is important to note that the data contained within these tables may offer supplementary insights into the clinical conditions of the patients that were not all extracted. We performed comprehensive result verification by comparing the row counts of the SQL queries against expected values and randomly inspecting sample data to ensure its quality and relevance.

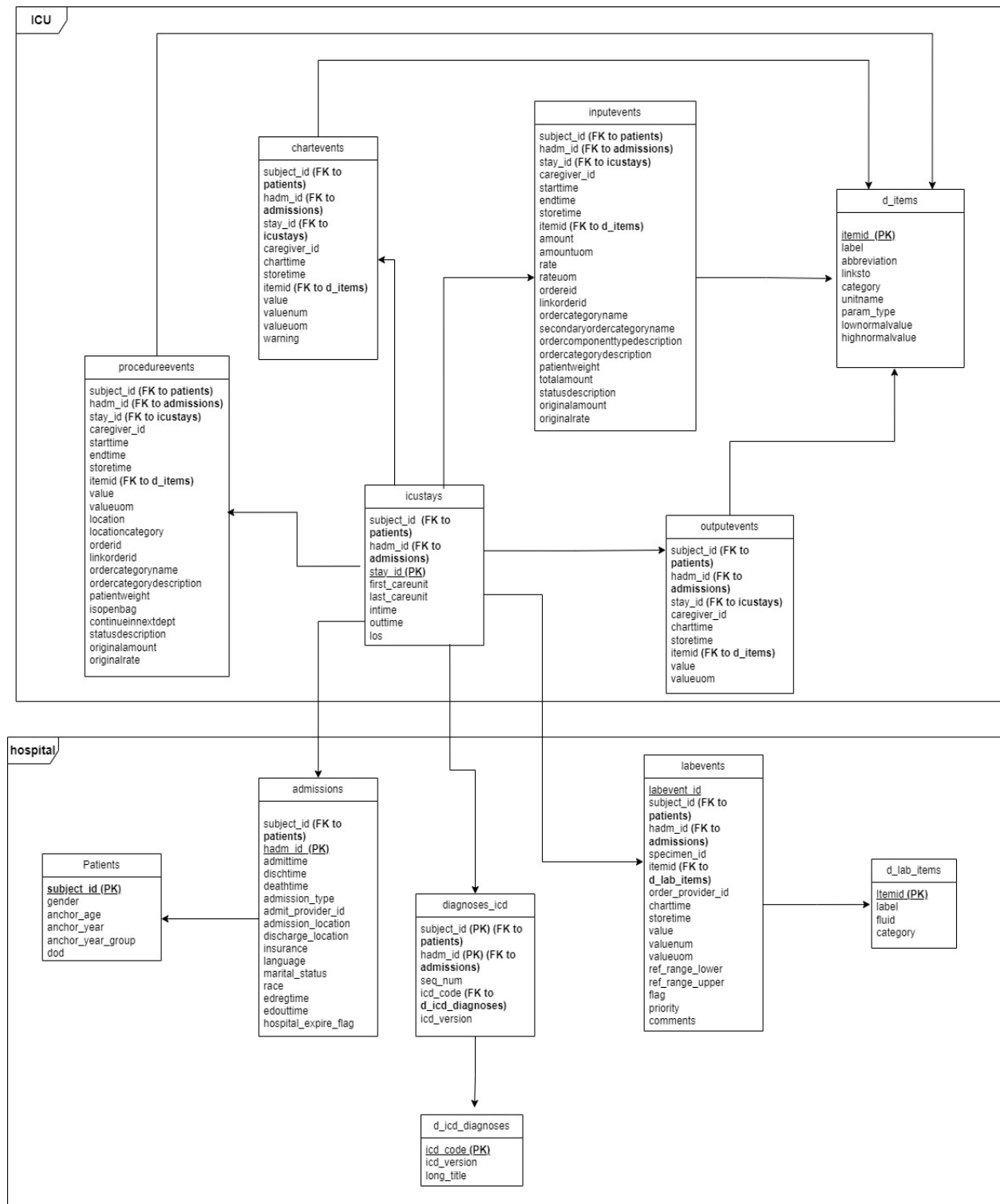


Figure 3-2: Relational schema of the tables used to extract the features and target values, guiding the formulation of the SQL queries for data extraction

3.1.3 Data preprocessing

Once we have established our model framework and collected our labeled dataset for the predictive task, it is essential to undertake data preprocessing to ensure its readiness for analysis. This step involves cleaning, transforming, and organizing the data to enhance its quality and suitability for the model, thereby facilitating more accurate predictions. LSTM models are known to be sensitive to variations in data quality and integrity. Given that much of the data in this study is raw, a thorough exploration of its quality for machine learning is a crucial step to enhance the reliability and consistency of our predictive model.

Missing values

In this study, missing data is managed using a hybrid approach that combined elements of ML-Remove, ML-Unique and ML-Traditional. This decision was driven by the need to balance dataset integrity with maintaining a sufficient sample size for robust model training. Specifically, missing values were replaced with a distinct category for categorical variables (ML-Unique). For example, a missing marital status was replaced by “*Unknown*”. Continuous variables were imputed using population medians (ML-Traditional) like the temperature set to the temperature site indicator. Furthermore, we developed binary indicators to account for missing data, enabling the model to consider the absence of a value without discarding the entire observation when a specific test was not performed on a patient. We used the median for this purpose, as it is less affected by extreme values compared to the mean. This approach relies on straightforward data transformation rather than complex algorithms. Any remaining missing values were completely removed (ML-Remove) to avoid introducing bias. This approach aimed to preserve as much data as possible while ensuring meaningful imputation where necessary, thereby minimizing potential biases in the ML developed for blood test frequency. While more detailed approaches to handling missing values could potentially yield better results [41], this was not within the scope of this study. Using a hybrid approach to handle missing values, balancing between preserving a substantial number of samples and addressing missing data issues, is a practical trade-off. This strategy acknowledges the importance of maintaining sample size for robust statistical analysis while also ensuring data integrity through appropriate imputation techniques.

We removed duplicate entries and treated values recorded within 5 minutes of each other as a single blood draw. This decision was based on the likelihood that these entries were either mistakenly entered multiple times by the nurse or represented two attempts due to an issue with the first draw. In either scenario, it suggested that only one valid blood draw should be considered.

Outliers

We identify and manage outliers in the dataset through a combined approach. Specifically, we employed Z-scores for features that exhibited a normal distribution and the Interquartile Range (IQR) method for those that were significantly skewed. Through visualization of the cleaned data, we assessed the impact of our outlier removal. This may involve reconsidering the thresholds applied in the Z-score and IQR methods to balance the removal of true outliers with the preservation of relevant data points. An important note here is that even if a value is statistically identified as an outlier, it may be clinically significant, particularly in a medical dataset. Thus, it's important to consider the context and consult clinical guidelines when making decisions about outliers in medical data.

Outliers in categorical data are identified through frequency analysis and visualization. Given that the MIMIC IV database has been thoroughly investigated, we recognize that some categories, while uncommon, may still be valid. Therefore, we will retain these categories to ensure our dataset accurately represents the true nature of ICU patients. Extended stays in the ICU are also regarded as outliers, so we will cap the frequency of blood tests at a maximum value after analyzing the distribution. Finally, intervals between blood draws that are significantly longer than typical are considered outliers, as they are rare and may introduce noise into our model, hindering its ability to predict more common intervals effectively. Such lengthy gaps may indicate patient stability and are generally less relevant in an intensive care context, where shorter intervals are more prevalent. Filtering out these outliers can enhance the dataset's relevance for model training. To determine the appropriate threshold for identifying these outliers, we will examine the distribution of intervals. This visualization will help establish a threshold that effectively removes outliers while retaining sufficient data for robust training.

Finally, based on the total number of blood draws per patient, an analysis is conducted to determine the maximum allowable output length. Any samples exceeding this length will be excluded and considered outliers. This adjustment is also based on the understanding that prolonged durations may indicate periods of clinical stability or a reduced need for predictive blood test frequency. Therefore, extending the planned blood test intervals beyond this threshold is deemed unnecessary for our analysis.

Encoding and Scaling Techniques

To effectively prepare the data for machine learning, we employed various encoding and scaling techniques:

- **One-Hot Encoding:** Categorical features, such as marital status or race, are transformed using one-hot encoding. This method converts categorical variables into a format that can be provided to machine learning algorithms, effectively eliminating any ordinal relationships that do not exist.
- **Word Embeddings for Diagnoses:** The primary diagnosis, represented as a series of words, will be processed using word embeddings. Word embeddings are a type of word representation that allows words to be represented as continuous vectors in a high-dimensional space. They capture the semantic meaning of words by placing words with similar meanings close together in this space. We will train a simple regex-based approach using Word2Vec model on tokenised description of diagnoses ICD.
- **Standardization of Numerical Features:** Numerical features will be scaled using standardization techniques such as Min-Max scaling. This ensures that all numerical inputs and outputs in our case are normalized to a common range, improving convergence during training and enhancing model performance.
- **Boolean Encoding for Gender:** Gender will be represented as a binary feature, allowing the model to utilize this information without introducing unnecessary complexity.

Feature Selection

Feature selection is a crucial preprocessing step in any machine learning pipeline, as it reduces dimensionality and enhances model performance. After conducting a preliminary exploration of relevant features based on domain knowledge and existing literature, we intend to implement machine learning feature selection techniques and assess feature correlation for further investigation and refinement. In this thesis, we combine filter, wrapper, and embedded feature selection methods to filter out unrepresentative features. In small scale dataset, it was found that ensemble feature selection by the intersection methods consistently perform better than wrapper method alone. Thus, using the findings of the authors in [42], *“if both classification accuracy and the feature reduction rate are considered to have the same level of importance, ensemble feature selection techniques using multi-intersection would be the better choice”*. The ensemble technique used in this thesis is:

- 1- Filter with SelectKBest() using Pearson correlation coefficient
- 2- Wrapper with RandomForestRegressor using Recursive Feature Elimination (RFE)
- 3- Embedded with Lasso regression

Each method utilizes a different model during data processing, mitigating the risk of overfitting when training the LSTM model. This approach accelerates the process by leveraging diverse models that provide complementary insights, effectively capturing different aspects of the data without redundancy. This minimizes the need for manual checks of each feature individually. We will use the most relevant features and intersection between them as an iterative process to determine the most informative features for this task. While the approach is strong, further additional techniques such as feature interaction and dimensionality reduction can be done to enhance the predictive power of our model but were not part of this analysis.

Proper feature structuring is vital to create a dataset that LSTM models can effectively process. We will ensure that the data is reshaped into appropriate sequences based on a one timesteps input of feature. By implementing these data quality measures and preprocessing steps, we aim to create a robust dataset that enhances the performance of our LSTM networks. This careful attention to data integrity will ultimately contribute to more accurate predictions of blood test frequencies in ICU settings, thereby supporting timely and informed clinical decisions.

Inconsistencies and Errors

We will conduct thorough checks for inconsistencies and errors within the dataset. This includes validating the format of dates and times, ensuring that numerical entries fall within logical ranges, and checking for duplicate records. Such validation is crucial for maintaining the quality of our dataset, as inconsistencies can lead to misleading predictions.

3.2 Model Configuration and Training

In this study, we aim to develop and explore LSTM networks for predicting blood test frequencies in the ICU. Initially, we will construct a simple LSTM model with a single hidden layer. This foundational model will allow us to understand the basic dynamics of the data and establish a baseline performance. Our feature selection from the previous step will help us refine this simple LSTM model and refine the feature set. On these feature sets, we will again train the model on Correntropy Loss to analyze its performance. To increase the model's performance, we will iteratively increase the complexity of the architecture by incorporating additional parameters, such as adding additional hidden layers, by using a GridSearch. We will implement early stopping to terminate training when validation performance begins to degrade, preventing overfitting. In addition to the LSTM models, we will train a multi-output linear regression model to serve as a baseline for comparison. This regression model will be critical during various stages of development, as it will help identify specific features that significantly influence predictions and inform necessary post-processing adjustments.

For this specific task, static patient data will act as the inputs to the model, while the sequential time series data for blood draws will serve as the outputs. By systematically experimenting with different LSTM configurations and comparing them against the baseline regression model, we aim to refine our approach and improve the accuracy of blood test frequency predictions in the ICU setting. Our approach will involve splitting the dataset into training, and validation using cross-validation (80%, 20% respectively) and test sets (80%, 20% respectively) to ensure robust model evaluation.

It is important to note that data augmentation was not implemented at this stage primarily because establishing a solid foundation with the correct features, loss function, and metrics is critical for this specific task. Without a well-defined set of features, any data augmentation could inadvertently introduce noise, leading to misleading results. Additionally, the chosen loss function must align with our objectives to ensure that the model learns effectively from the data. We believe in an iterative approach where refining our model's core components is the priority. Once we have established a robust baseline, we can then explore data augmentation techniques to enhance model performance. This strategic approach ensures that any improvements made are meaningful and contribute positively to the model's outcomes.

3.3 Tools and Technologies

The pipeline employs a combination of tools and technologies, including SQL for data extraction, and Python for modeling and analysis, using libraries such as scikit-learn, TensorFlow, pandas, NumPy, matplotlib, and seaborn. Excel is primarily used for analysis and organization. Additionally, we regularly shuffle the data to ensure it appears more realistic and freer from bias. This approach helps to enhance the model's robustness and validity.

3.4 Conclusion

In this chapter, we outlined the complete methodology employed to develop a predictive model for determining blood test frequencies in ICU patients using LSTM. Our main objective is to create a robust machine learning model capable of making timely and accurate predictions at the time of patient admission to the ICU. We began by defining the essential features and target values for our analysis, drawing from a range of clinical data including demographics, vital signs, and laboratory test results. The features were selected based on existing literature, emphasizing those known to influence blood test intervals. We employed the MIMIC-IV database as our primary data source, ensuring the dataset's quality and relevance for our predictive task. With additional feature selection techniques, we aim to define the appropriate set of features. Data preprocessing is a crucial step, involving a hybrid approach to handle missing values and ensure the integrity of the dataset. This careful preparation allows us to mitigate biases and maintain a sufficient sample size

for robust model training. We also discussed the importance of data quality, highlighting the techniques we would implement to check for outliers and inconsistencies, which are particularly significant for LSTM models since they are sensitive to data variations.

For model selection and training, we laid out a structured approach, starting with a simple LSTM model and progressively increasing complexity by incorporating advanced architectural elements. This systematic enhancement aims to improve prediction performance while maintaining model robustness. We also plan to establish a baseline using a multi-output linear regression model, which will facilitate comparisons and help identify key features influencing our predictions. Through this structured approach, we aim to contribute valuable insights into blood test frequency predictions in ICU settings, ultimately enhancing patient care by informing timely clinical decisions.

CHAPTER 4 RESULTS

In this section, we present the results of our investigation into predicting blood test frequencies upon ICU admission using LSTM. We will provide a comprehensive overview of the dataset utilized, including the results of relevant features, preprocessing steps, training methodologies, and loss functions acquired. The models will be evaluated and compared using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2), and a hybrid metric, with detailed performance analyses provided for each model.

4.1 Data Preparation Outcomes

In this section, we outline how we constructed our labeled data, along with the limitations and thresholds discussed earlier. Our goal is to provide a detailed account of the extraction and preprocessing steps taken.

4.1.1 Results of Data Extraction from MIMIC-IV

The features (Input X) and targets (output t_0 to $t_{\text{discharge}}$) were all extracted from MIMIC-IV. To do so, we first aimed to obtain all laboratory test results conducted at the time of admission to an ICU ward for each patient, making “*charttime*” the defining attribute for our query. Specifically, we extracted laboratory tests that were closest to the “*intime*” attribute. Careful attention was given to handling duplicates to ensure that only the first occurrence of each lab event within the specified timeframe of approximately six hours post-admission is selected. The query in Figure 4-1 begins by creating a Common Table Expression (CTE) named “*DistinctLabEvents*”, which includes a selection of relevant fields such as “*subject_id*”, “*hadm_id*”, “*stay_id*”, “*charttime*”, and test values from the joined tables. We use the “*ROW_NUMBER()*” function to assign a unique rank (*row_num*) to each lab event per patient stay, ordering them by their proximity to the “*intime*” timestamp. This allows us to identify the laboratory test closest to admission and to retain only the first occurrence of each “*charttime*” within each group, eliminating any duplicate or redundant entries. This was essential for maintaining the integrity of the data, especially when dealing with time-sensitive information like blood sample collection times. In the subsequent “*SELECT*” statement, we retrieve only those records where *row_num* = 1, ensuring that we include only the first lab event for each patient within the specified timeframe. The results are then ordered by

“*stay_id*” and “*charttime*” for clarity. This query extracts the features for laboratory test results detailed in the previous chapter where each laboratory test has its own id, refer to Appendix A - Table A. 1 for more information on their types and units.

```
WITH DistinctLabEvents AS
(
  SELECT icu.subject_id, icu.hadm_id, icu.stay_id,
         le.charttime, icu.intime, icu.outtime, le.value,
         le.valuenum, le.valueuom,
         ROW_NUMBER() OVER
         (
           PARTITION BY icu.subject_id, icu.hadm_id, icu.stay_id
           ORDER BY ABS (TIMESTAMP_DIFF(le.charttime, icu.intime, SECOND)) ASC
         ) AS row_num
  FROM `physionet-data.mimiciv_icu.icustays` AS icu
  JOIN `physionet-data.mimiciv_hosp.labevents` AS le
    ON   icu.subject_id = le.subject_id
    AND   icu.hadm_id = le.hadm_id
  WHERE le.specimen_id IS NOT NULL
    AND   le.charttime BETWEEN TIMESTAMP_SUB(icu.intime, INTERVAL 6 HOUR)
    AND   TIMESTAMP_ADD(icu.intime, INTERVAL 6 HOUR)
    AND   le.itemid = 50983
)
SELECT subject_id, hadm_id, stay_id, charttime, intime, outtime, value, valuenum, valueuom
FROM DistinctLabEvents
WHERE row_num = 1
ORDER BY stay_id, charttime;
```

Figure 4-1: SQL query for laboratory tests values from the MIMIC-IV database, i.e. le.itemid = 50983 corresponds to sodium (Na)

Similarly, for vital signs, we established a Common Table Expression (CTE) named “*RankedEvents*”, Figure 4-2. This CTE retrieves all relevant fields from the “*chartevents*” table and assigns a row number (row_num) to each vital sign measurement based on the “*charttime*” for each “*stay_id*”. We joined the “*chartevents*” table with the “*icustays*” table on “*stay_id*” to capture vital signs recorded as close to admission time as possible. Rather than aggregating or calculating the mean, we selected only the first vital measurements to preserve the raw data, in contrast to other studies that aggregated vital signs based on proximity to “*intime*” [13]. Again, refer to Appendix A - Table A. 1 for all ids of vital signs. The queries for weight, race, gender and other medical information like primary diagnoses were also extracted in a similar fashion using the corresponding tables. Refer to the MIMIC documentation for more details³.

³ Medical Information Mart for Intensive Care, Freely available medical data for research, [MIMIC \(mit.edu\)](https://mimic.mit.edu/)

```

WITH RankedEvents AS
(
  SELECT ce.*,
         ROW_NUMBER() OVER (PARTITION BY ce.stay_id ORDER BY ce.charttime)
         AS row_num
  FROM `physionet-data.mimiciv_icu.chartevents` ce
  JOIN `physionet-data.mimiciv_icu.icustays` icu ON ce.stay_id = icu.stay_id
  WHERE ce.itemid = 220045
        AND ce.charttime BETWEEN icu.intime AND icu.outtime
)
SELECT ce.*
FROM   RankedEvents ce
WHERE  ce.row_num = 1

```

Figure 4-2: SQL query for vital signs, i.e. ce.itemid = 220045 corresponds to the first measure of the heart rate of the patient at admission time to an ICU ward

It is important to note that in MIMIC-IV, the "*charttime*" recorded in the *labevents* table reflects when the test results were documented, rather than the time when the blood sample was drawn. This discrepancy between sampling time and charting time can introduce variability for the model. However, for a preliminary analysis, it is important to note that we used raw information directly using "*charttime*" to establish a baseline before refining the analysis with more sophisticated methods. Furthermore, we did not distinguish whether the measurements came from Point-Of-Care Testing (POCT) or a laboratory which would also introduce variability in the frequency. Using raw "*charttime*" can still yield valuable insights to identify general trends or patterns in blood test frequencies and can serve strong foundation for future work. Therefore, for the target values (t_0 to $t_{\text{discharge}}$), the query used returned a list of unique ICU stays where blood-related lab measurements were taken. Each row represents an ICU stay with the time of the lab measurement, patient ID, admission ID, stay ID, and the ICU stay's start and end times sorted by the "*charttime*". According to the MIMIC-IV database documentation, different "*specimen_ids*" with the same "*charttime*" likely represent the same blood sample. Therefore, the query focused on grouping by "*charttime*". This was confirmed when comparing the counts of distinct "*charttime*" values with counts of unique "*specimen_ids*", revealing a pattern of multiple specimens per sample. In fact, a single blood sample can be divided into multiple specimens for various tests, and different lab tests might be recorded with different "*specimen_ids*" but collected simultaneously. Since our aim was to

retrieve the timing of a single blood sample, we achieved this by using “*charttime*” in the following query Figure 4-3.

```
SELECT DISTINCT icu.subject_id, icu.hadm_id, icu.stay_id, lab.charttime,
                icu.intime, icu.outtime
FROM `physionet-data.mimiciv_icu.icustays` AS icu
JOIN `physionet-data.mimiciv_hosp.labevents` AS lab ON icu.hadm_id = lab.hadm_id
JOIN `physionet-data.mimiciv_hosp.d_labitems` AS dlab_item ON lab.itemid = dlab_item.itemid
WHERE
    dlab_item.fluid = 'Blood'
    AND lab.charttime >= icu.intime
    AND lab.charttime <= icu.outtime
    AND icu.los < 50
GROUP BY icu.subject_id, icu.hadm_id, icu.stay_id, lab.charttime,
         icu.intime, icu.outtime
```

Figure 4-3: SQL Query to capture blood test times representing individual draw events

Table 4-1 presents an example of blood sample collection data. In this particular ICU case, it was noted that 11 blood samples were collected, with their corresponding timings recorded in the “*charttime*” column. As part of our data cleaning procedure mentioned earlier, we established a 5 minute interval ($5/60 = 0.08333$) around each “*charttime*” entry. Any intervals shorter than 5 minutes were identified and excluded, as they were likely indicative of human error or duplication. Additionally, it is essential to note that although “*subject_id*” may occur multiple times, each instance signifies a unique datapoint.

Table 4-1: Example of recorded timings for blood samples collected between patient admission and discharge from an ICU ward

subject_id	hadm_id	stay_id	charttime	intime	outtime
18106347	24305596	30588857	2110-01-11T10:22:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T11:05:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T11:50:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T13:00:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T16:07:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T19:00:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-11T19:14:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-12T01:29:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-12T01:52:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-12T09:08:00	2110-01-11T10:16:06	2110-01-12T17:17:47
18106347	24305596	30588857	2110-01-12T13:35:00	2110-01-11T10:16:06	2110-01-12T17:17:47

We further validated the query by comparing the number of blood draws with the total recorded timings, confirming they were equal. The query can be found in Appendix B Figure B.1. For all the extracted data (features and target values), we randomly selected a few “*subject_id*” values and manually reviewed their hospitalization timelines. This process confirmed the accurate extraction of data from the MIMIC-IV database. Given that this dataset adheres to professional and globally recognized healthcare protocols and standards, we can confidently assert that the data originates from a reliable and professional source. Therefore, there was no necessity for creating synthetic data or conducting data simulations.

4.1.2 Outcomes of Data Preprocessing

In total, we had 71,184 unique patients, although some may have been readmitted and were counted as separate entries due to different *hadm_ids* or *stay_ids*. Each patient is accompanied by their corresponding vital signs, laboratory tests, and clinical information prior to preprocessing. Most patients had 50 blood draws or fewer, as shown in Figure 4-4. Patients with more were classified as outliers to simplify the initial structure and establish a more manageable foundation.

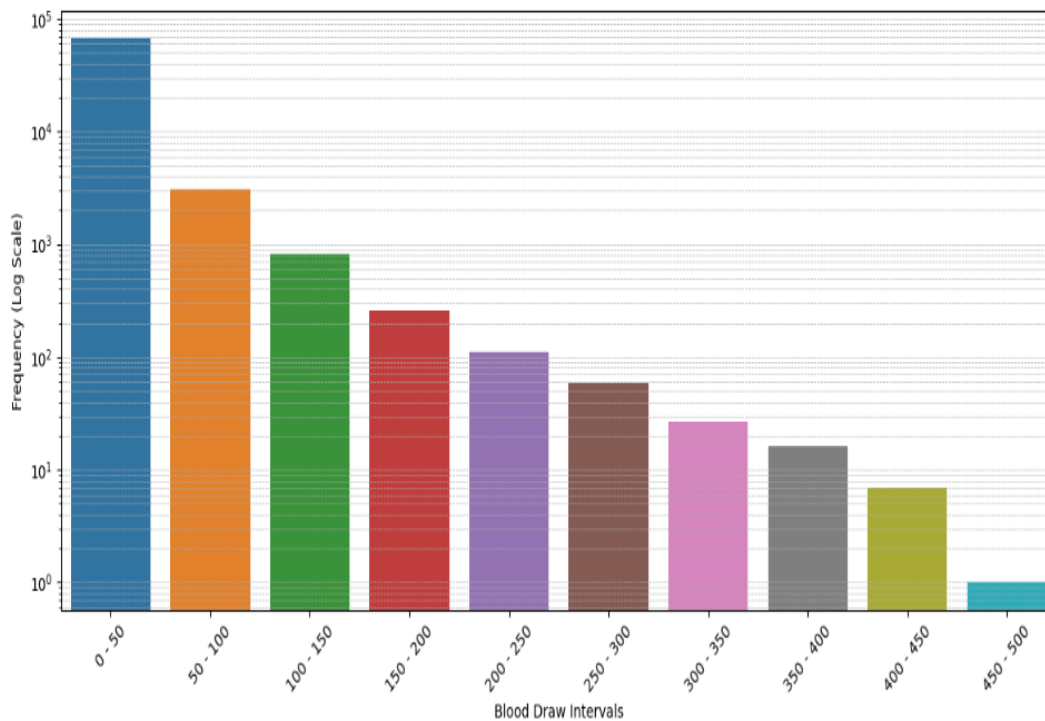


Figure 4-4: Number of patients categorized by blood draw frequency

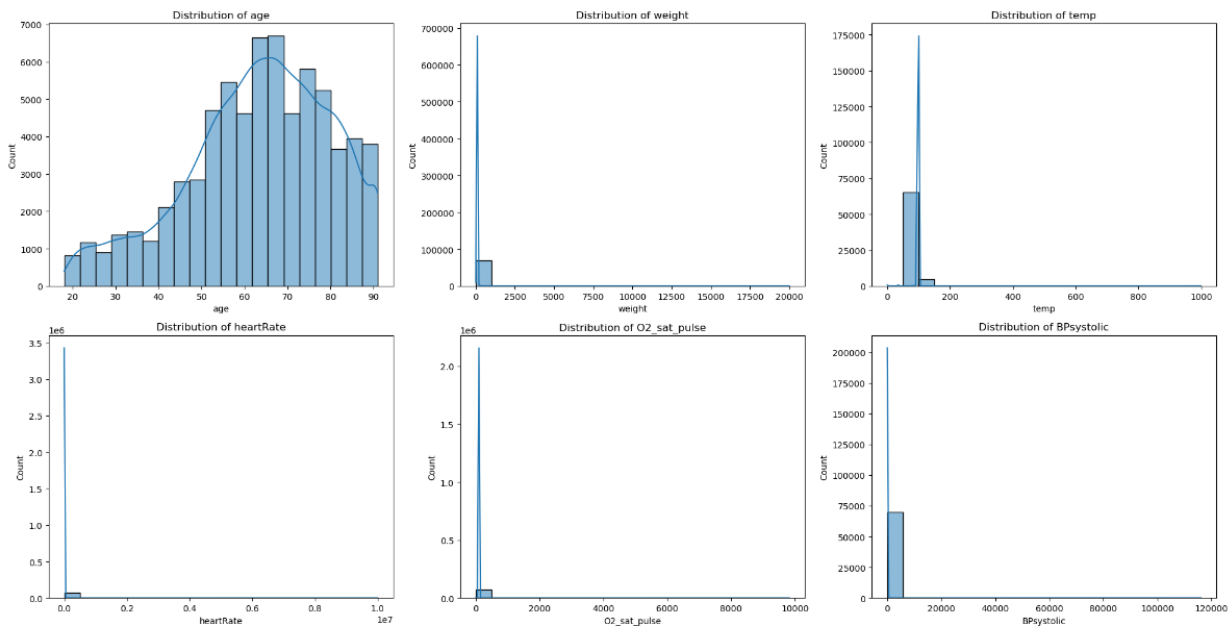
Missing values

Missing values were most prevalent in laboratory tests, which is expected since vital signs and clinical information are typically recorded more frequently. We observed that blood HCO_3 , calcium, and phosphate had the highest number of missing values, with approximately 12,000 samples. This is an important consideration for feature selection in a realistic environment. Following these, MCH, MCV, RBC, and RDW showed about 6,000 missing values each. Additionally, WBC, hemoglobin, platelet count, hematocrit, potassium, creatinine, and chloride had nearly 5,000 missing values. The only demographic information that was frequently missing was marital status. To retain a substantial amount of data and make it more representative of real-world scenarios, the missing values were replaced with a boolean flag and the median as mentioned in the previous section.

Outliers

For numerical features, we visualized the data by creating histograms and density plots to examine their distributions Figure 4-5 - a. As we analyzed each feature, we assessed the skewness to

determine the type of distribution. For features with skewness less than 0.5, indicating an approximately normal distribution, we calculated the Z-scores and filtered out rows with absolute Z-scores exceeding the specified threshold. For features with significant skewness, we computed the IQR and identified outliers based on the established bounds. After filtering, we counted the number of outliers removed for each feature and recorded this information in a dictionary to evaluate the extent of outlier removal. We observed a considerable loss of information in the creatinine feature, prompting us to apply specific conditions for handling this feature. Subsequently, we removed the identified outliers from the dataset and re-visualized the cleaned data with histograms and density plots to assess the effect of our outlier removal, which now almost all exhibited a Gaussian distribution Figure 4-5 – b. The distribution for the rest of the numerical features is in Appendix B - Figure B.4. The code for this function is presented in APPENDIX C - C. 1.



a)

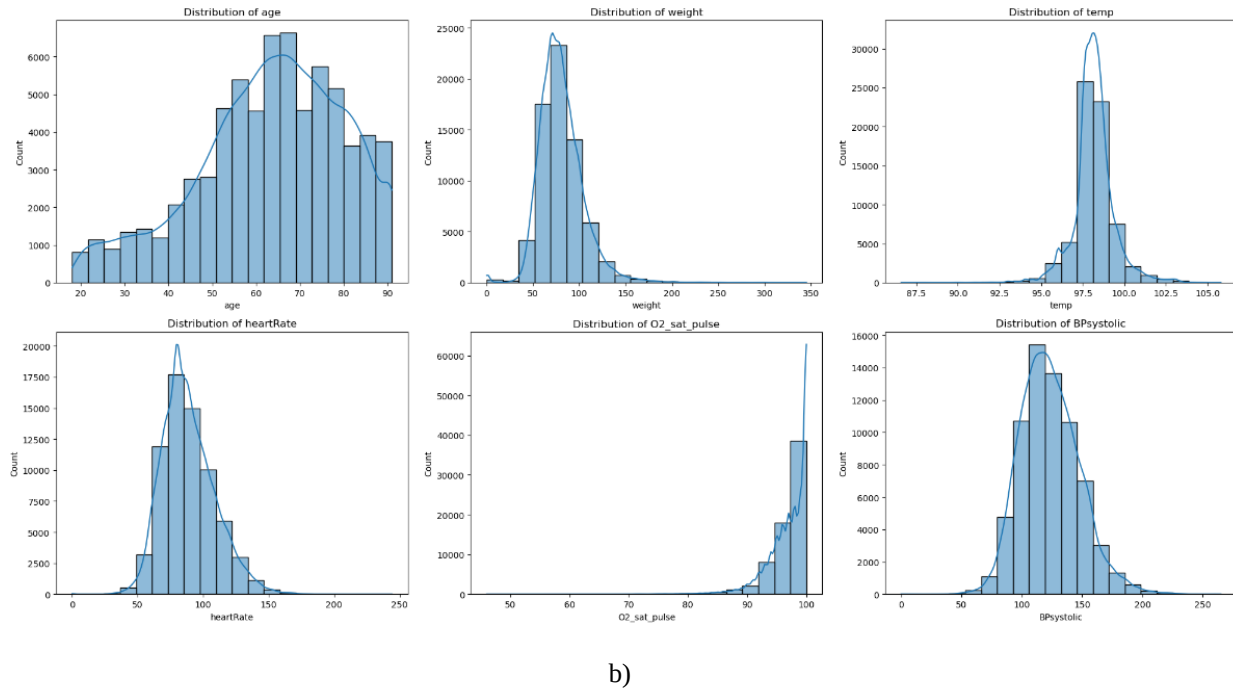


Figure 4-5: Visualization of numerical features - a) before applying outlier detection Z-score and IQR - b) after applying Z-score and IQR

Encoding and Scaling Techniques

For categorical values, primary diagnoses were embedded using a simple regex-based Word2Vec model with a vector size of 100 to capture their semantic relationships. The model was implemented as follows: `Word2Vec(sentences=tokenized_descriptions, vector_size=100, window=5, min_count=1, sg=1)`, applied to the tokenized descriptions of the primary diagnoses. See Appendix C - C. 2 for the complete code.

The total size of the dataset after preprocessing was 62,980 samples, consisting of 149 features and 50 target values, with values ranging from 0 to 24 hours. The rationale was that beyond 24 hours, patients are typically viewed as being in a more stable condition, making a predictive model for blood test frequency less critical since the urgency for frequent testing diminishes and patient management strategies may shift towards less intensive monitoring. Refer to APPENDIX B - Figure B.2 for the distribution of blood draws that were > 24 hours.

Feature selection

The correlation shown in Table 4-2, along with the heatmap in Figure 4-6, highlights significant relationships among numerical features that are essential for informing feature selection in machine learning models. Notably, strong correlations (≥ 0.7) such as between BP diastolic and BP mean (0.93) and between hemoglobin and RBC (0.91) suggest that these features are closely related and may convey similar information while training, potentially leading to redundancy in the model. The moderate correlation between urea nitrogen and creatinine (0.65) highlights their collective relevance as kidney function indicators, while the strong correlation between sodium and chloride (0.70) suggests they may behave similarly in terms of physiological regulation. Overall, while these correlations can provide valuable insights in clinical settings, they necessitate careful consideration to avoid multicollinearity issues when training machine learning models [43]. Therefore, strategic feature selection based on these correlations could enhance model performance and interpretability, ensuring that the most informative and non-redundant features are utilized for predicting outcomes effectively.

Table 4-2: Correlation table of the significant relationship

Numerical Feature	Numerical Feature	Level of Correlation
BPsystolic	BPmean	0.75
BPsystolic	BPdiastolic	0.58
BP diastolic	BPmean	0.93
Sodium	Chloride	0.70
UreaNitrogen	Creatinine	0.65
Hemoglobin	RBC	0.91
Hemoglobin	Hematocrit	0.94
Hematocrit	RBC	0.91
MCHC	MCH	0.51
MCV	MCH	0.82

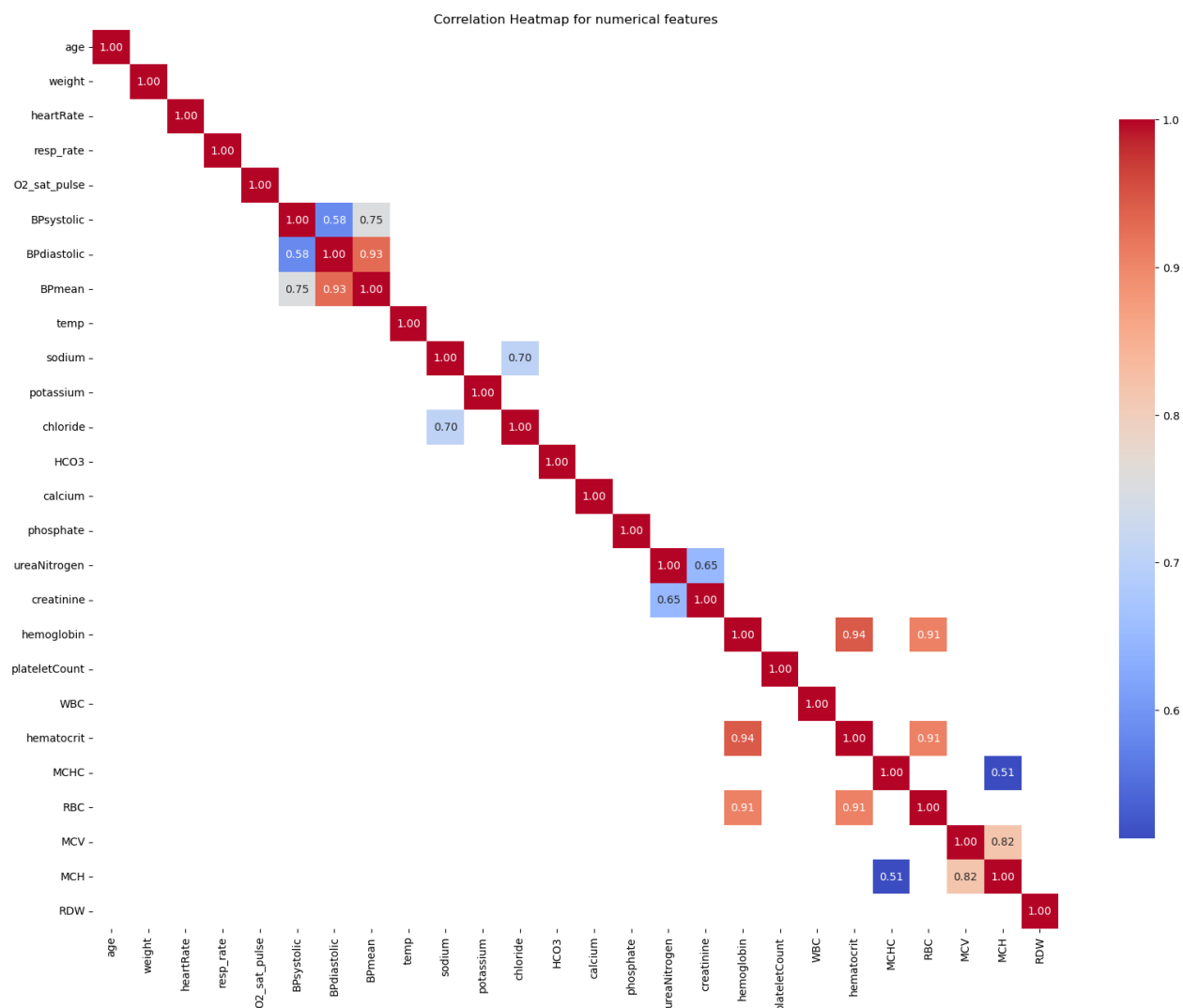


Figure 4-6: Correlation heatmap for numerical features including vital signs, blood test results and clinical patient data > 0.5

Similarly, in evaluating the inclusion of features such as `Marital_status_unknown` and `insurance_unknown`, alongside the perfectly inversely correlated features `insurance_other` and `insurance_medicare`, several considerations emerged regarding their impact on the ML model, see Appendix B - Figure B.3. The moderate correlation of approximately 0.6 between marital status

and insurance status suggests that these features may convey similar information, potentially introducing noise if a significant portion of the dataset is categorized as “*unknown*”. Conversely, the perfect negative correlation (-1) between the insurance categories indicates their mutual exclusivity, yet again raising concerns about multicollinearity if both features are retained. To inform the decision-making process, we conducted experiments by building models with and without these features, evaluating their predictive performance. Furthermore, if “*unknown*” categories are prevalent in the data, options for handling them could include imputation or the creation of a distinct category to capture this information without excessive data loss. Overall, careful evaluation and empirical testing will guide the determination of whether to retain these features for model improvement or exclude them for greater simplicity and clarity or vice-versa.

The rationale behind using feature selection techniques was to identify which features influenced the target variable, specifically the frequencies of blood samples (from t_0 to $t_{\text{discharge}}$). By systematically evaluating the influence of each feature, we aimed to simplify the decision-making process for retaining or removing features, rather than relying on time-consuming manual selection and numerous combinations. This approach not only accelerates the workflow but also provides a clearer overview of the features that are valuable for inclusion in our initial analysis. We applied an ensemble of methods similar to the one presented by the authors in [44], filter, wrapper and embedded. The features that were most and least frequently selected across all three methods are listed in the following Table 4-3. Based on this table, there were no top features that intersected across all methods. However, we did observe a clear intersection among the least favored features, with 8 features overlapping, see Figure 4-7. The features included 'race_BLACK', 'insurance_Medicare', 'gender', 'race_OTHER', 'insurance_Medicaid', 'insurance_Other', 'marital_status_SINGLE', and 'marital_status_MARRIED'.

Table 4-3: Top 20 and bottom 20 features chosen by each feature selection method

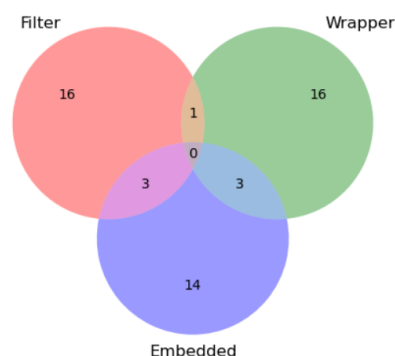
Method	Top 20 Selected Features	Least 20 Selected Features
Filter method using Pearson correlation with SelectKBest()	1. emb_26 2. emb_27 3. emb_80 4. WBC 5. WBC_missing 6. MCHC_missing	1. marital_status_DIVORCED 2. MCH 3. race_OTHER 4. marital_status_SINGLE 5. race_BLACK 6. marital_status_MARRIED

	7. RDW_missing 8. MCH_missing 9. MCV_missing 10. RBC_missing 11. hematocrit_missing 12. hemoglobin_missing 13. plateletCount_missing 14. UreaNitrogen_missing 15. HCO3_missing 16. creatinine_missing 17. chloride_missing 18. first_careunit_Cardiac Vascular Intensive Care Unit (CVICU) 19. first_careunit_Neuro Intermediate 20. first_careunit_Neuro Surgical Intensive Care Unit (Neuro SICU)	7. insurance_Medicaid 8. insurance_Other 9. insurance_Medicare 10. gender 11. emb_2 12. emb_29 13. emb_39 14. emb_43 15. emb_55 16. emb_61 17. emb_66 18. emb_67 19. emb_81 20. emb_91
Wrapper Multioutput RandomForestRegressor using RFE (taken the mean of the weights on the feature on all targets)	1. temp 2. heartRate 3. plateletCount 4. WBC 5. BPsystolic 6. resp_rate 7. phosphate 8. RDW 9. age 10. ureaNitrogen 11. MCHC 12. BPdiastolic 13. potassium 14. creatinine 15. chloride 16. sodium 17. MCH 18. BPmean 19. RBC 20. hemoglobin	1. insurance_Medicaid 2. insurance_Medicare 3. insurance_Other 4. marital_status 5. race_BLACK race_OTHER 6. race_WHITE 7. MCHC_missing 8. WBC_missing 9. hemoglobin_missing 10. plateletCount_missing 11. hematocrit_missing 12. HCO3_missing 13. ureaNitrogen_missing 14. creatinine_missing 15. chloride_missing 16. gender 17. MCH_missing 18. MCV_missing 19. RBC_missing 20. RDW_missing
Embedded Method Lasso	1. temp 2. weight 3. RDW 4. Phosphate 5. emb_3 6. emb_14 7. emb_22 8. emb_26 9. emb_32 10. emb_33 11. emb_34 12. emb_45 13. emb_70 14. emb_90	1. MCH 2. marital_status_WIDOWED 3. race_UNKNOWN 4. RBC 5. emb_31 6. emb_81 7. first_careunit_Coronary Care Unit (CCU) 8. hemoglobin_missing 9. MCHC_missing 10. insurance_Medicaid 11. race_WHITE 12. insurance_Other 13. RDW_missing

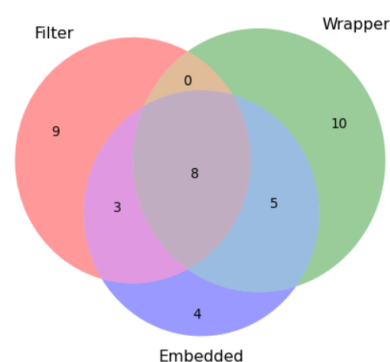
	15. emb_93 16. emb_94 17. MCV 18. ureaNitrogen 19. first_careunit_Neuro Intermediate 20. first_careunit_Neuro Surgical Intensive Care Unit (Neuro SICU)	14. marital_status_DIVORCED 15. gender 16. race_OTHER 17. race_BLACK 18. marital_status_SINGLE 19. insurance_Medicare 20. marital_status_MARRIED
--	--	---

Venn Diagramm of the Top 20 Features From Different Selection Methods

Venn Diagramm of the Bottom 20 Features From Different Selection Methods



a)



b)

Figure 4-7: Venn diagramm of the – a) top 20 features and b) bottom 20 features – in different selection methods

Given that no features from the top 20 were included, we extended the analysis to the top 50 features, Figure 4-8. Eleven features demonstrated consistent significance across all three methods: ['WBC', 'first_careunit_Cardiac Vascular Intensive Care Unit (CVICU)', 'BPsystolic', 'calcium', 'resp_rate', 'ureaNitrogen', 'heartRate', 'emb_22', 'RDW', 'phosphate', 'hemoglobin']. In contrast, twelve features were found to have the least impact on the models: ['insurance_Medicare', 'marital_status_WIDOWED', 'marital_status_SINGLE', 'insurance_Other', 'marital_status_MARRIED', 'race_BLACK', 'marital_status_DIVORCED', 'emb_7', 'race_WHITE', 'race_OTHER', 'insurance_Medicaid', 'gender']. Notably, gender appeared among the least influential features, contradicting some literature that suggests it enhances model performance. We will assess the impact of including and excluding gender on the LSTM models. These techniques clearly indicate that demographic features, such as insurance, marital status, and race, had minimal contribution to the model. We will also investigate the potential benefits of removing correlated features, testing each one individually to identify which provides a better fit.

In both scenarios, we will evaluate performance with and without these features, aiming for a “subset of features that are highly correlated with the response variable yet uncorrelated to each other” [43] that can increase the model’s capability.

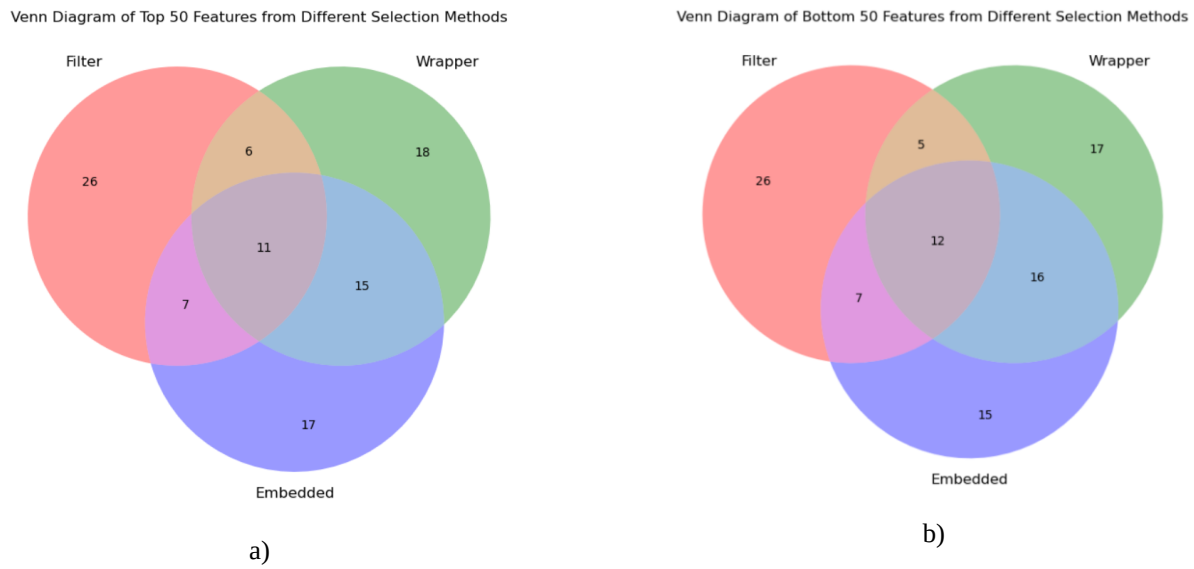


Figure 4-8: Venn diagram of the top 50 – a) and bottom 50 features – b) in different selection methods

4.2 Model Training Results

Training a machine learning model is an iterative and experimental process that requires careful consideration of feature selection and performance evaluation. The first step involved evaluating the multi-output linear regression model to assess its performance using all available features. We obtained an overall Mean Absolute Error (MAE) of 1.33 hours and a Mean Squared Error (MSE) of 3.58 hours. For sequence lengths, the overall MAE was 12.06 timestamps, and the overall MSE was 18.98 timestamps, resulting in a combined metric of 6.69 and an overall performance of 39.12%, indicating a poor prediction performance, as shown in Figure 4-9 - a. This scatter plot displays the true versus predicted sequence lengths with a red dotted line as a benchmark, indicating that for the model to be considered effective, all points should align closely with this line. The value of 3 served as a scale for evaluation, allowing us to assess how accurately the data was predicted in terms of sequence length, given the task of predicting variable sequences.

Following the analysis of the feature selection presented earlier, the removal of the least significant features from the model yielded overall MAE and RMSE values of 1.32 and 3.57 hours, respectively, with sequence lengths averaging 11.87 timestamps and 18.76 timestamps, resulting in a combined metric of 6.63, Figure 4-9 - b. The difference between the MAE and RMSE indicates the presence of outliers in the dataset, which is typical of raw medical data. Furthermore, the increase in performance by almost 5% indicates that the feature selection technique was effective. However, we observed that not all features are correlated with the target values, suggesting that there may be missing information that the model is not capturing. To address this issue, we added a timing-related feature - specifically, the timing of the first blood draw - into the model. We think that this enhancement allows the model to capture more nuanced patterns within the data and help the LSTM with temporal information. Notably, the feature “*time_interval_t0*” proved to be significant, enhancing the performance of a theoretical multioutput linear regression model. Furthermore, we were able to exclude calcium and chloride without sacrificing performance. In the missing value and correlation analysis along with the feature selection, these features consistently emerged as strong candidates for removal. Figure 4-9 – c illustrates the improved outcomes, with a prediction accuracy of 45.85%. The average MAE and MSE are now 1.31 and 3.58, respectively, while the sequence lengths are 11.54 and 18.43, leading to a combined metric of 6.43. Our goal is for this combined metric to approach 0, which would indicate that both the values and the sequence lengths are predicted accurately.

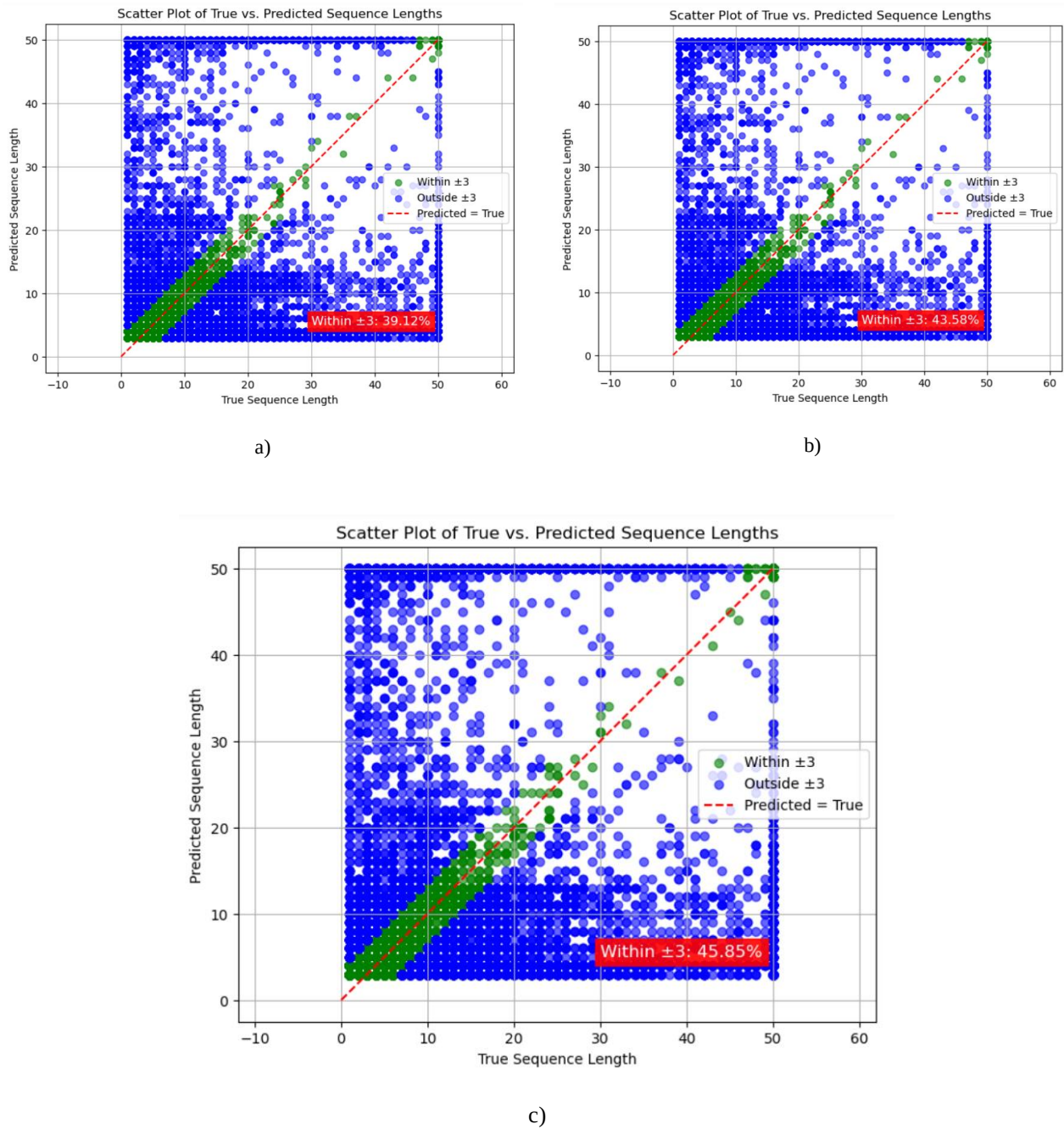


Figure 4-9: MultiOutput Linear Regression – a) using all features b) removed least significant features from Ensemble Method c) using *time_interval_t0* as additional feature generated better result, removing calcium, and chloride based on correlation and testing with or without the features

4.2.1 Iterative training Process

Given that most medical data exhibit non-linear characteristics, they extend beyond the capabilities of linear regression. The previous tests established a baseline for training our LSTM model and highlighted the value of incorporating a time-related feature that had not been considered during the literature review. This underscores the significance of feature selection in addressing specific problems. In this approach, we defined and compiled a single LSTM model, specifying the architecture with one hidden layer (256 neurons), an output layer (to generate a value for each timestamp) and the loss function using MSE. We proceeded with using a single train-test split (80%, 20%) along with a validation split during training (80%, 20%). We reshaped the training and test data to the appropriate format for the LSTM model, which requires input in the shape of (samples, time_steps, features) which resulted in (samples, 1, features) for our input and (sample, 50 – t_0 to $t_{\text{discharge}}$ (padding with 0), 1) for the output layer. Setting up early stopping helps monitor validation loss during training, preventing overfitting by halting the process if the validation loss does not improve over a specified number of epochs. The code is detailed in the Appendix C - C. 3 - Simple LSTM model. While this method is straightforward and effective, it may not provide as robust an estimate of model performance as k-fold cross-validation, potentially leading to overfitting if the model learns patterns specific to the validation split. Therefore, we evaluate the model on a separate test dataset that was not used during training to obtain a true measure of performance. This approach is particularly suitable for our exploratory phase, as it simplifies the process and allows for quicker results. However, for more comprehensive evaluations, especially with larger datasets, we might consider k-fold cross-validation in the future.

The series of figures from Figure 4-10 to Figure 4-13 illustrate the results on the test set when training the single LSTM model with all features included. In contrast, the figures from Figure 4-14 to Figure 4-17 display the test set results for the LSTM model trained using only the selected features. Both the values and the sequence length were crucial and required accurate predictions. Therefore, we aimed for a clear visualization of the model's performance regarding these aspects. For example, in Figure 4-10, we can see that the model did not predict any sequence exceeding 20 timestamps. A histogram can effectively illustrate that by providing the distribution of timestamps comparing both the actual and predicted sequence lengths Figure 4-11. At the same time, when

evaluating the values at each timestamp, a box plot, Figure 4-12, can be effective to determine if the data exhibits any outliers, as well as to visualize the distribution of values, including the median, quartiles, and overall variability. In both MAE and RMSE, we can see variations of the values at different timesteps, highlighting significant fluctuations in the predicted values compared to the true values. To complement these metrics, R^2 remains a useful indicator of how effectively our model captures variance in the data, even when the underlying relationship is non-linear. The negative R^2 value showed in Figure 4-12 indicates potential issues, such as inadequate feature selection, model simplicity, noise in the data, or possible overfitting to the training set. This is further highlighted by presenting the histogram for each timestamp, Figure 4-14. We observed a modest increase in performance when training the model on the selected features.

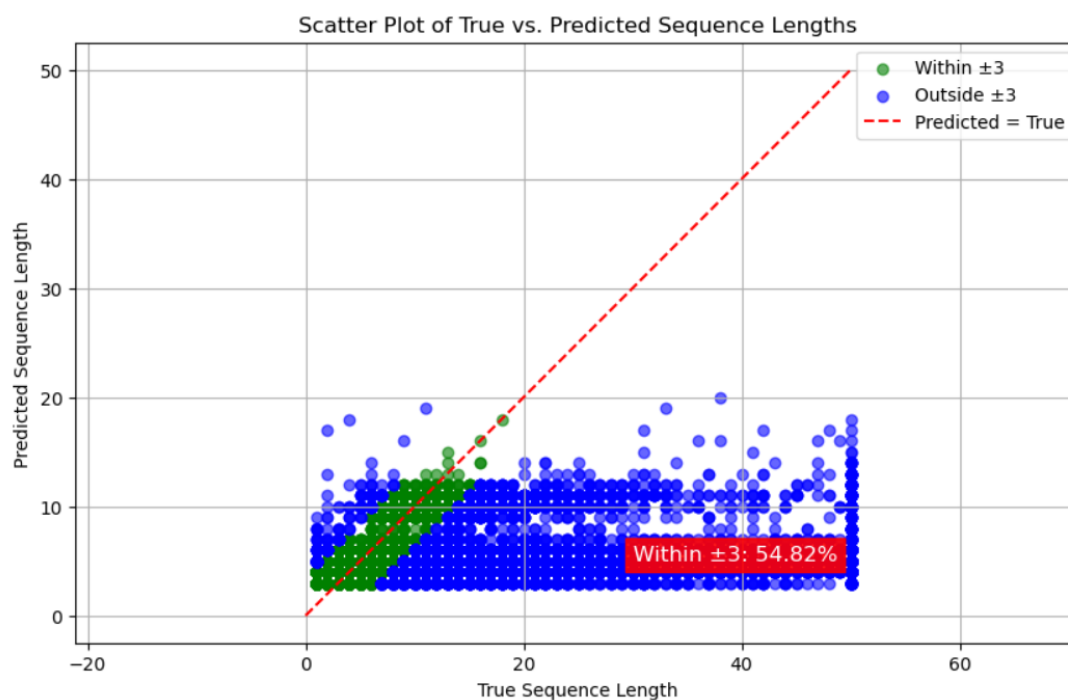


Figure 4-10: Single LSTM trained on all features - Scatter plot of the true vs. predicted sequence length using the red dotted line as benchmark

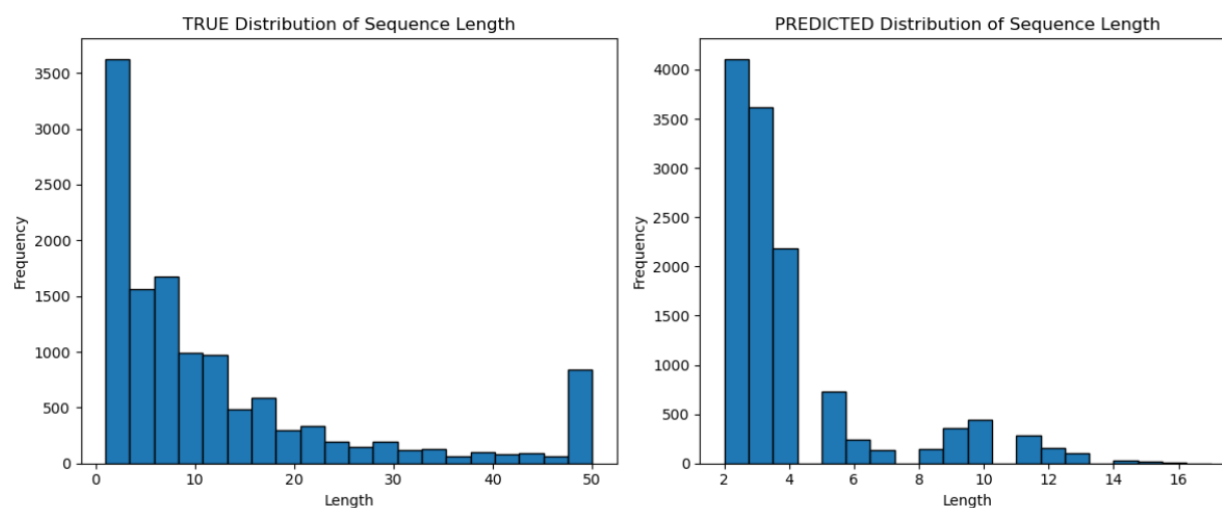


Figure 4-11: Single LSTM trained on all features - Distribution of timestamps comparing both the actual and predicted sequence lengths

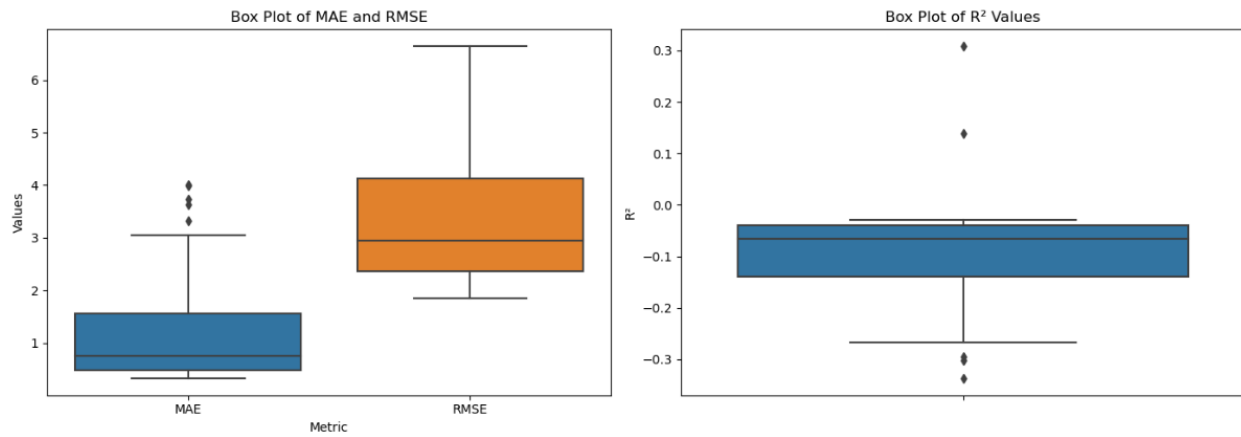


Figure 4-12: Single LSTM trained on all features - Box plot of MAE and RMSE for the value in each timestamp alongside R²

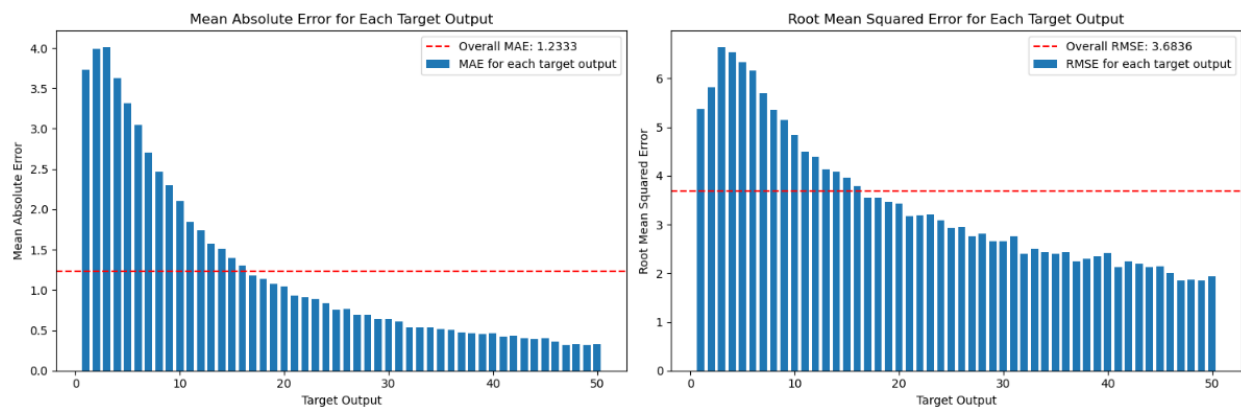


Figure 4-13: Single LSTM trained on all features – MAE and RMSE for each timestamp t_0 to t_{50} with the overall red dotted line

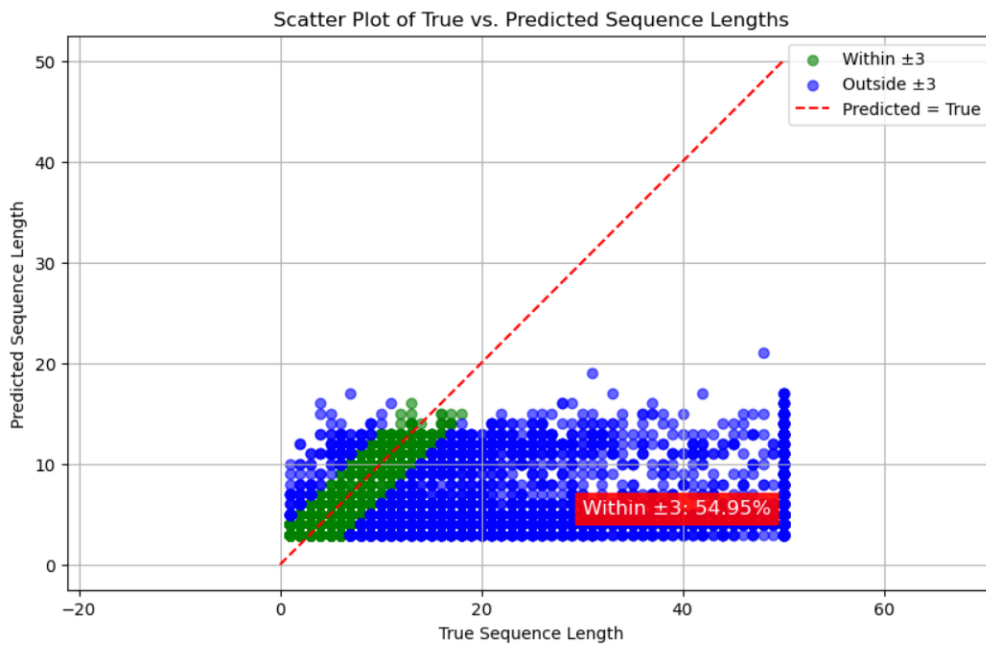


Figure 4-14: Single LSTM trained on selected features - Scatter plot of the true vs. predicted sequence length using the red dotted line as benchmark

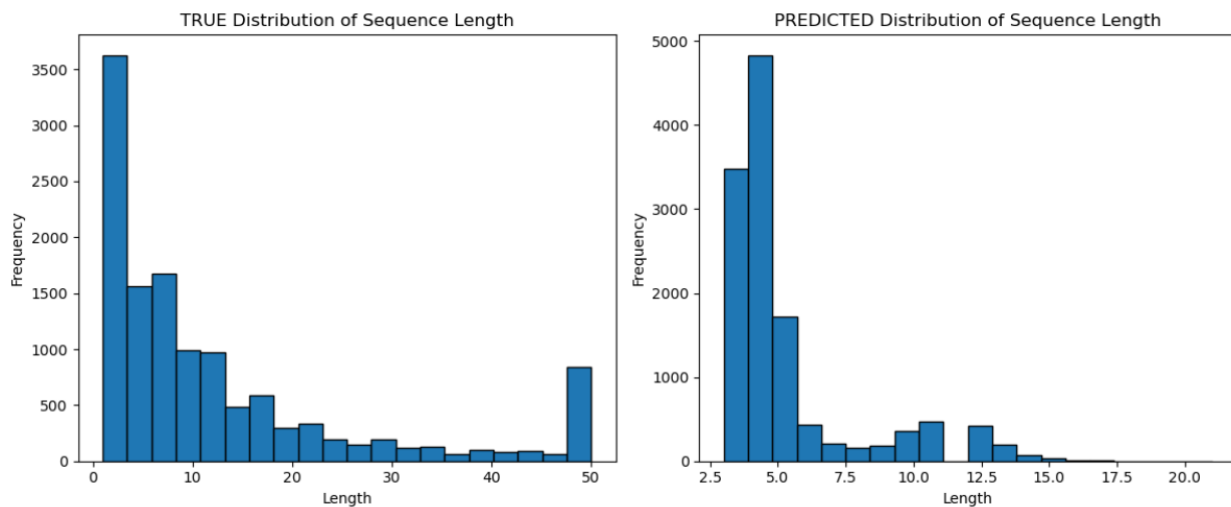


Figure 4-15: Single LSTM trained on selected features - Distribution of timestamps comparing both the actual and predicted sequence lengths

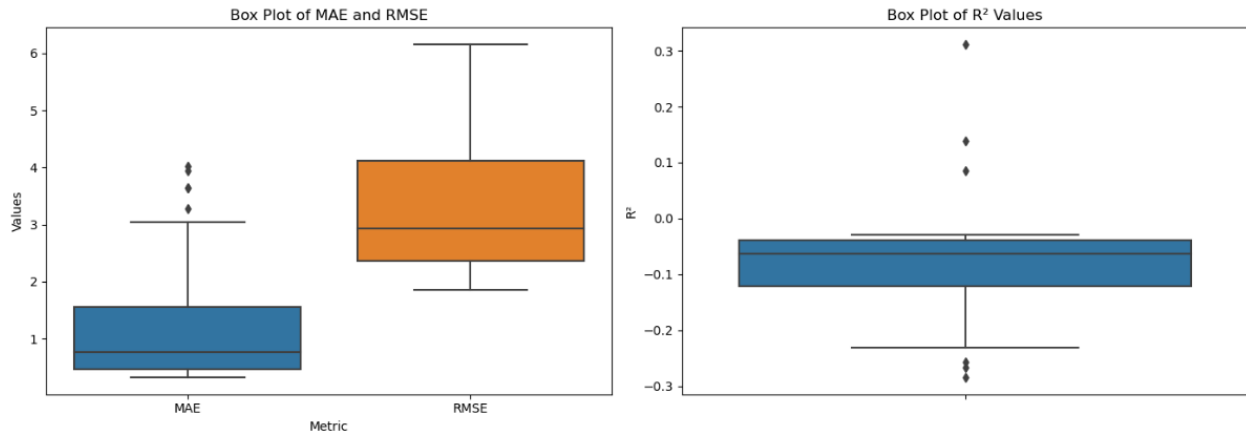


Figure 4-16: Single LSTM trained on selected features - Box plot of MAE and RMSE for the value in each timestamp alongside R^2

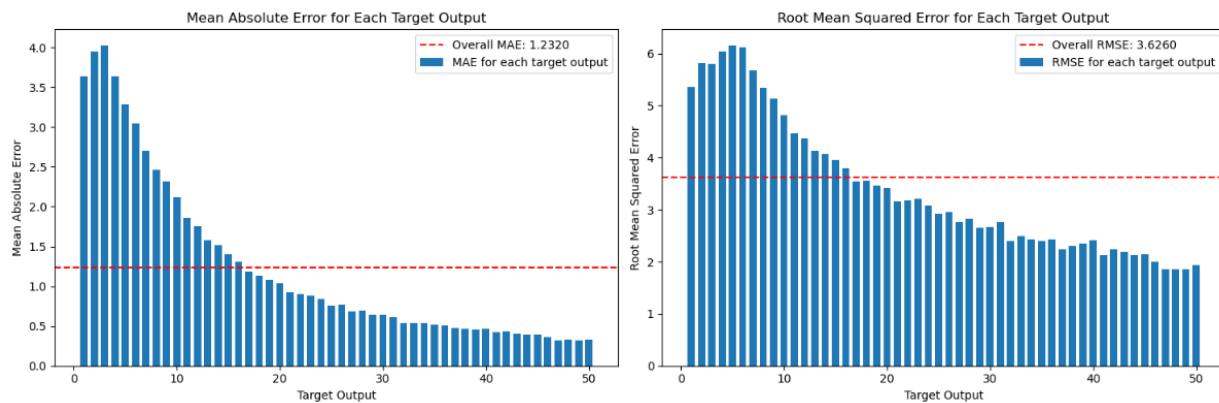


Figure 4-17: Single LSTM trained on selected features – MAE and RMSE for each timestamp to t_{50} with the overall red dotted line

Observably, the model struggled to accurately identify the appropriate stopping points at the timestamps, failing to predict the end-of-sequence token of zeros used for padding. Although we experimented with an alternative token, such as -1, this approach resulted in increased Mean Absolute Error (MAE) and Mean Squared Error (MSE). To address this issue, we employed a pragmatic heuristic, which we then fine-tuned to enhance performance. The mechanism involved truncating the sequence when a pattern of five consecutive decreases in predictions was observed. Specifically, we monitored each prediction sequence and counted instances where the current value

was less than the previous one. If this decrease count reached the threshold of five - tuned for optimal performance - we reset all values from the second-to-last detected decrease onward to zero, effectively truncating the sequence. We incorporated this mechanism because we noticed that, rather than generating the EOS token, the LSTM was producing a downward trend after a certain timestamp, prompting us to truncate the sequence at that point, see Figure 4-18. In the subsequent section, we evaluated an additional metric to account for outliers in the data and the variability in errors we observed. Furthermore, we increased the model's complexity through hyperparameter tuning.

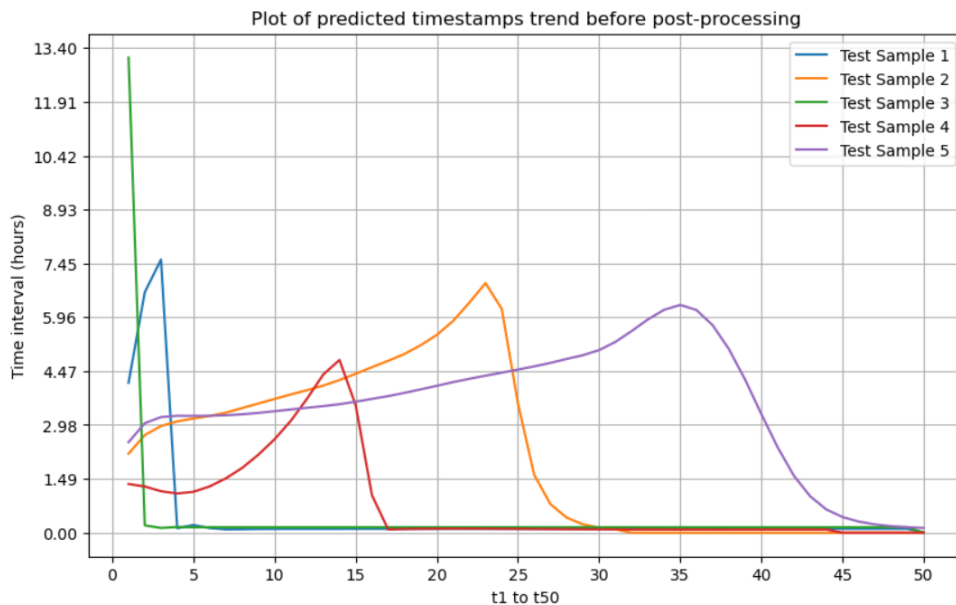


Figure 4-18: Trend of the predicted values at each timestamp using 5 samples from the test set

4.2.2 Analysis of Loss Function Performance

By experimenting with different loss functions, we can guide our model to focus on specific characteristics of the data, potentially leading to better performance. We experimented with Correntropy loss [39] used in traffic flow prediction as it effectively addresses non-Gaussian noise in the data. Similarly, in medical data, we need to account for specific types of noise to train the model effectively and focus on relevant aspects. We developed a function to optimize the sigma

parameter, aiming to identify the best value for the task, and compared its performance against MSE. The range of the kernel size tested were those employed in [39] but we added [5,7,9,10] to search for the optimal kernel size (σ). The results are presented in the following Table 4-4 and further visualized independently in Figure 4-19. Notably, the Correntropy Loss had a more pronounced effect on the sequence length (total number of timestamps) and minimal impact on the values at each timestamp. A kernel size of 1 yielded the highest percentage of accurately predicted sequence lengths, though it resulted in a slight increase in the combined metric compared to $\sigma = 2$ and $\sigma = 7$. Given our goal to prevent patients from developing iatrogenic anemia and to provide accurate predictions for blood sampling frequency, we chose $\sigma = 1$, even though it introduces slight timing deviations that can be adjusted through other parameters. An MAE of approximately ± 1.23 hours is relatively minor compared to the risk associated with ± 15 draws, especially since we understand that by the fourth draw, patients may already be at risk of developing anemia.

Table 4-4: Performance metrics of various kernel size (σ)

Sigma	MAE	MSE	SL MAE	SL MSE	Combined	Percentage ± 3
0.1	1.23	3.78	9.17	15.64	5.2	52.34
0.2	1.22	3.71	8.85	15.24	5.03	54.36
0.5	1.22	3.63	8.72	15.06	4.97	54.97
1	1.23	3.62	8.69	15.03	4.96	55.28
2	1.23	3.62	8.61	14.91	4.92	54.92
3	1.23	3.62	8.65	14.96	4.94	54.93
5	1.24	3.62	8.64	14.94	4.93	54.74
7	1.23	3.62	8.58	14.83	4.9	54.66
9	1.23	3.63	8.61	14.86	4.92	54.06
10	1.24	3.63	8.71	15.06	4.98	54.4

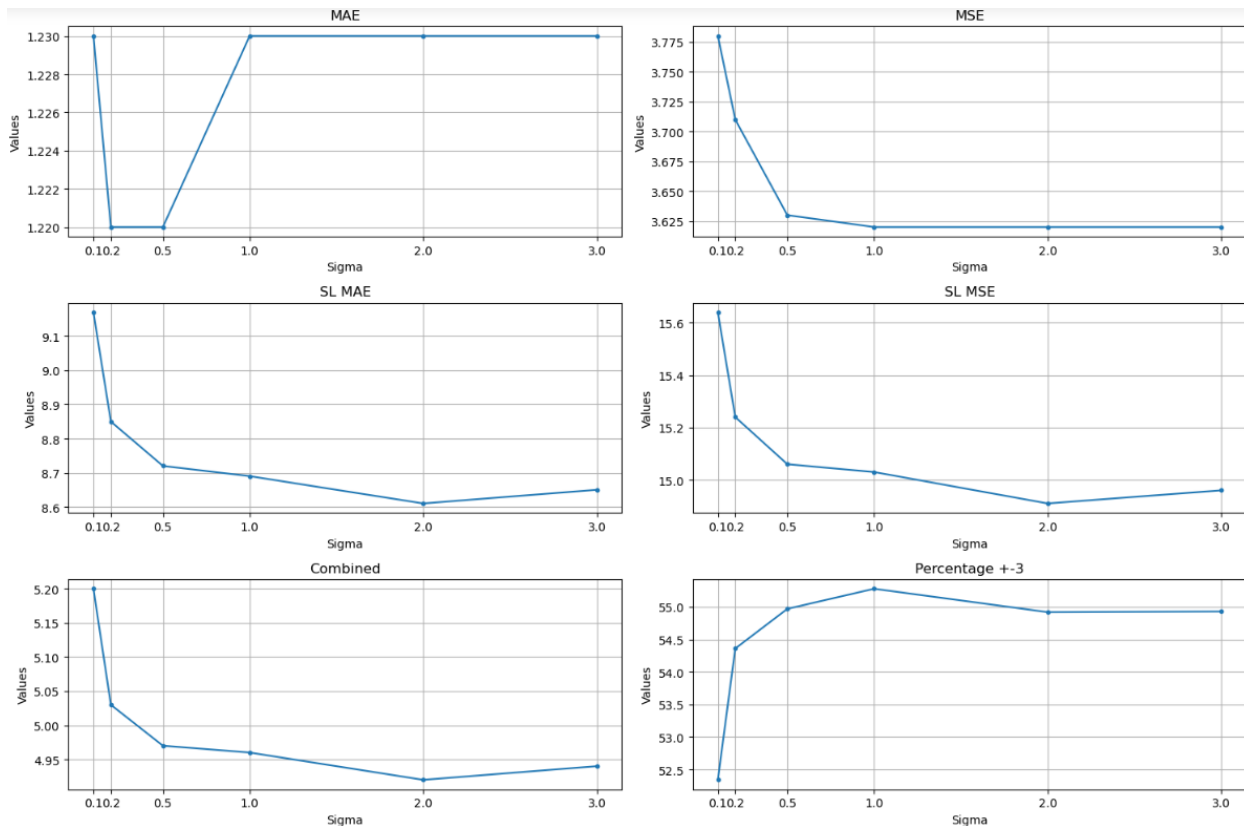


Figure 4-19: Visualization of the performance metrics across various kernel size (σ)

4.3 LSTM Improvement through Hyperparameter Tuning

Hyperparameter tuning was conducted using GridSearch to optimize various parameters, including the number of hidden layers, units within each layer, batch size, and dropout rate. The parameter grid consisted of values for the number of layers (2, 3, and 4), initial units (256 and 512), dropout rates (0.2 and 0.5), and batch sizes (512 and 64), resulting in a total of 24 combinations. To enhance model performance, we implemented early stopping with a maximum of 50 epochs and a patience of 3, monitoring validation loss based on mean squared error (MSE). The results of the tuning process are detailed in Appendix A - Table A. 2. The optimal configuration consisted of four hidden layers with dropout rates of 0.5 applied after each layer except the last one, featuring 512, 256, 128 and 50 units respectively, see Appendix C - C. 4 for the final model configuration. We set $\sigma = 1.0$ for the Correntropy loss function. Refer to Appendix B - Figure B.5 for an overview of the training/validation loss. Our model achieved an MAE of 1.22 hours and an RMSE of 3.62 hours,

with SL MAE and RMSE of 8.67 and 15.01 timestamps, respectively. Overall, we reached a 55.18% accuracy within ± 3 in sequence length $\pm$ MAE. The "*relu*" activation function was utilized in all tests.

After conducting these experiments and modifying the loss function while increasing the model's complexity to enhance its capacity for capturing relevant information, we found that the features still lacked certain critical details. To address this, we trained the model with an additional feature: the number of blood draws per patient. This was particularly relevant after noting only marginal improvements in performance during the Grid Search, where enhancements were primarily observed in the sequence length rather than in the values at each timestamp. Incorporating this feature provided important context, suggesting that patients with more blood draws may exhibit different patterns or behaviors compared to those with fewer. This additional context significantly improved the model's predictions and overall performance. We compared the results of this additional feature by training the tuned LSTM with MSE and Correntropy Loss, with visualization of the performance displayed in the series of figures, Figure 4-20 to Figure 4-29. The distribution of predicted sequence lengths closely mirrored the true distribution Figure 4-22, resulting in significant reductions in MAE and MSE Figure 4-20, now falling below one hour for MAE and 2.7 hours for RMSE. The most noteworthy improvements were observed in SL MAE and SL RMSE, now recorded at 2.02 and 3.33 timestamps, respectively when using MSE as the loss function. When using Correntropy Loss, the results showed further significant improvements, and in both cases, R^2 values stabilized, suggesting that the model is now effectively fitting the data. Figure 4-28 effectively illustrates how the predicted blood sampling frequency closely aligns with the actual blood frequency for each patient, providing a standardized approach. However, we must remain cautious of potential overfitting.

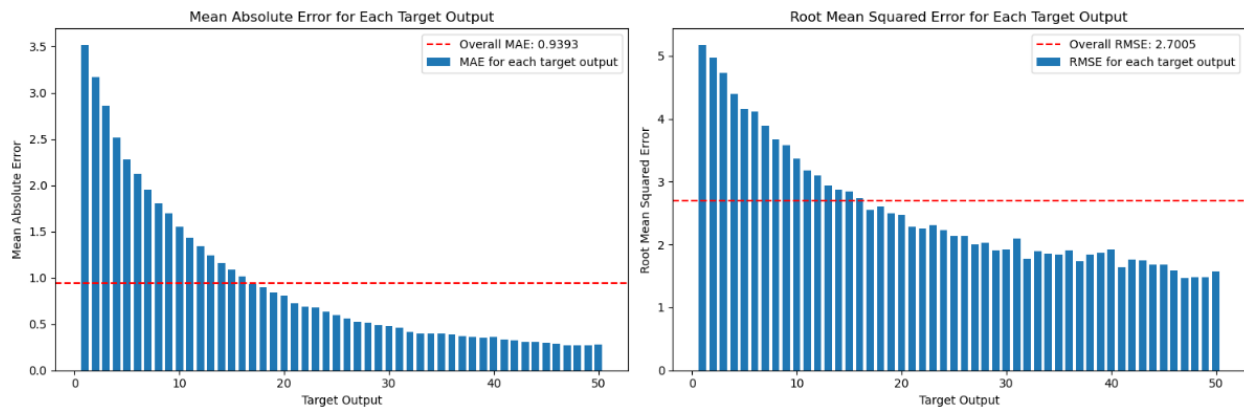


Figure 4-20: MAE and RMSE for each timestamp t_0 to t_{50} with tuned LSTM using selected features, gender, time_interval_ t_0 and number of blood draw - MSE Loss

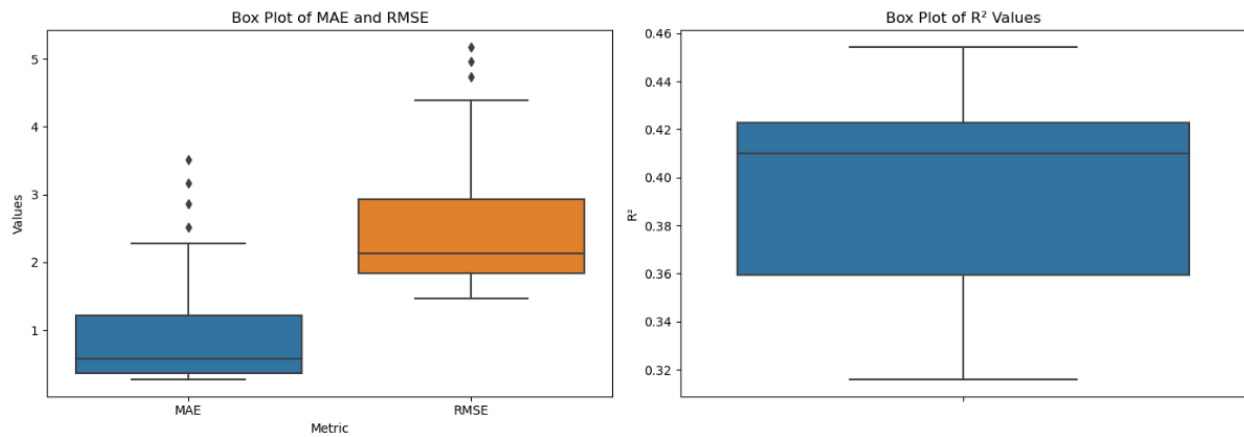


Figure 4-21: Box plot of MAE and RMSE for the value in each timestamp alongside R^2 with tuned LSTM using selected features, gender, time_interval_ t_0 and number of blood draw - MSE Loss

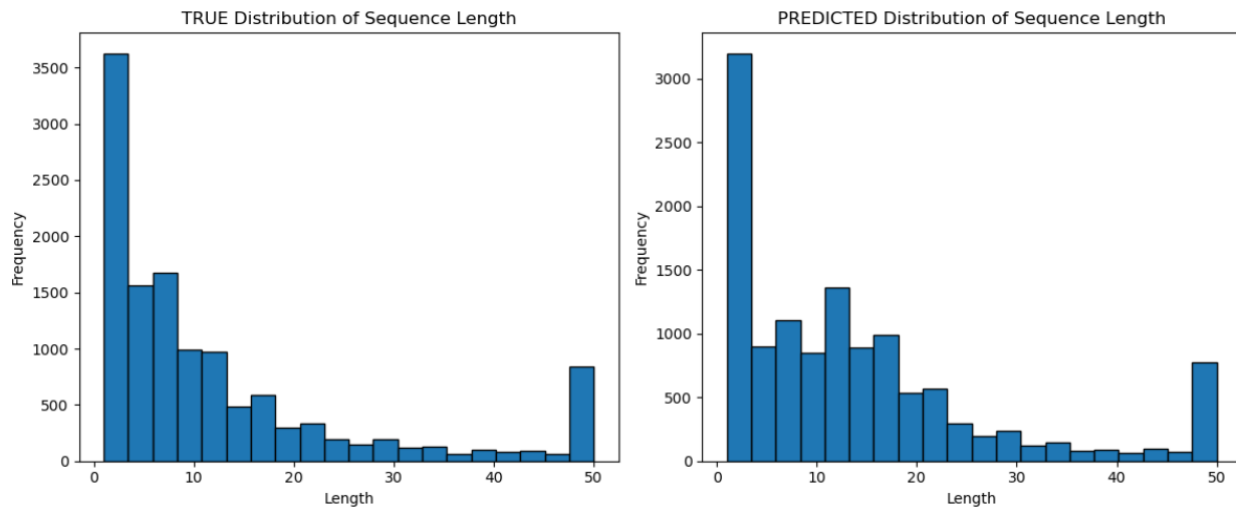


Figure 4-22: Distribution of timestamps of actual and predicted sequence with tuned LSTM using selected features, gender, time_interval_t0 and number of blood draw - MSE Loss

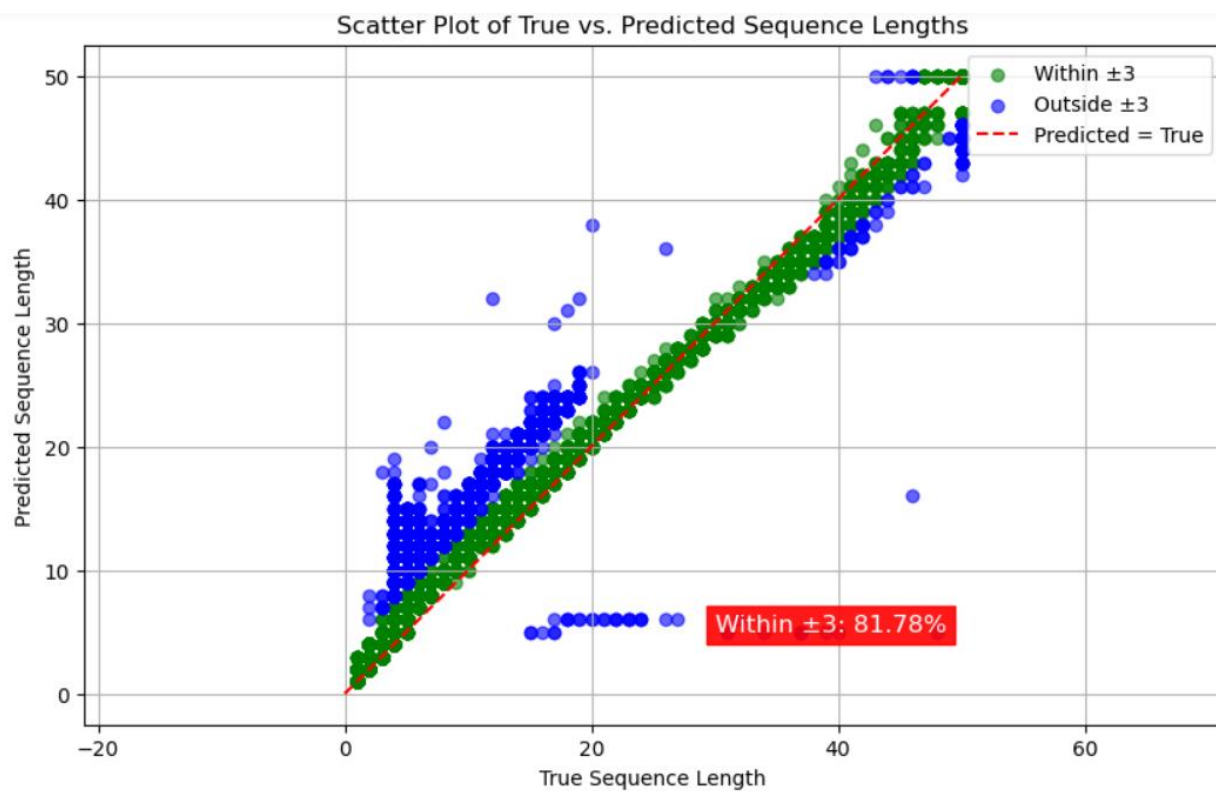


Figure 4-23: Scatter plot of the true vs. predicted sequence length with tuned LSTM using selected features, gender, time_interval_t0 and number of blood draw - MSE Loss

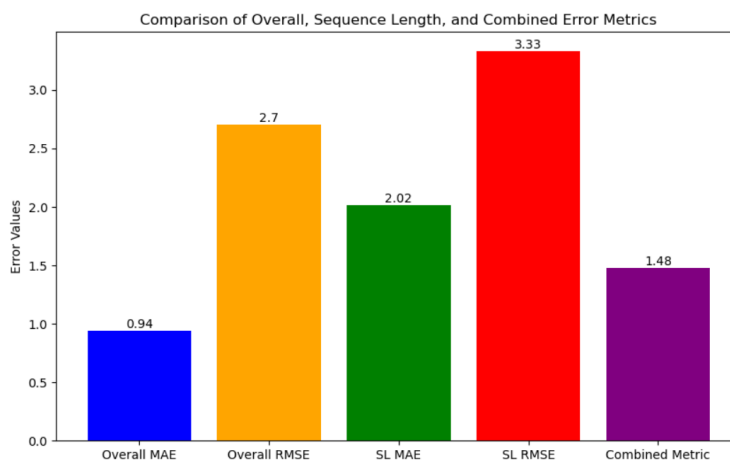


Figure 4-24: Evaluation metrics with tuned LSTM using selected features, gender, time_interval_t₀ and number of blood draw - MSE Loss

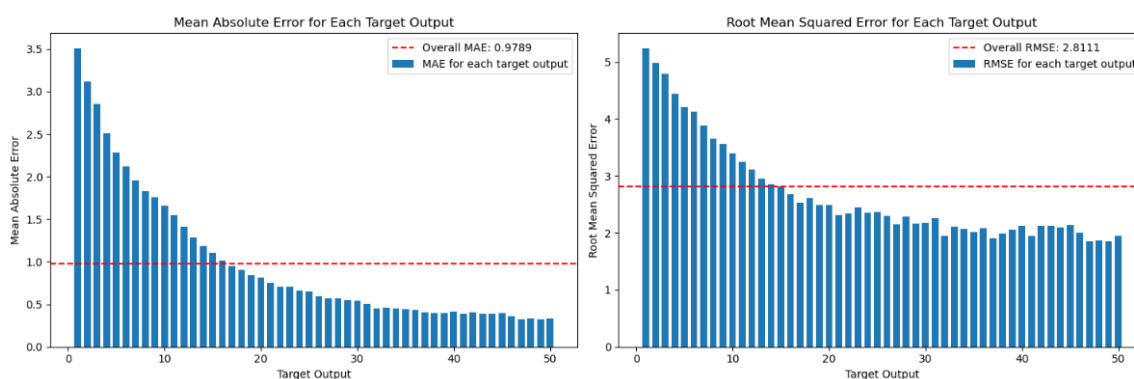


Figure 4-25: MAE and RMSE for each timestamp t₀ to t₅₀ with tuned LSTM using selected features, gender, time_interval_t₀ and number of blood draw - Correntropy Loss

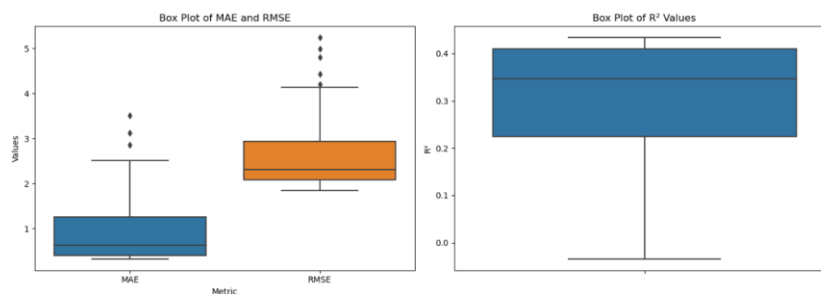


Figure 4-26: Box plot of MAE and RMSE for the value in each timestamp alongside R² with tuned LSTM using selected features, gender, time_interval_t₀ and number of blood draw - Correntropy Loss

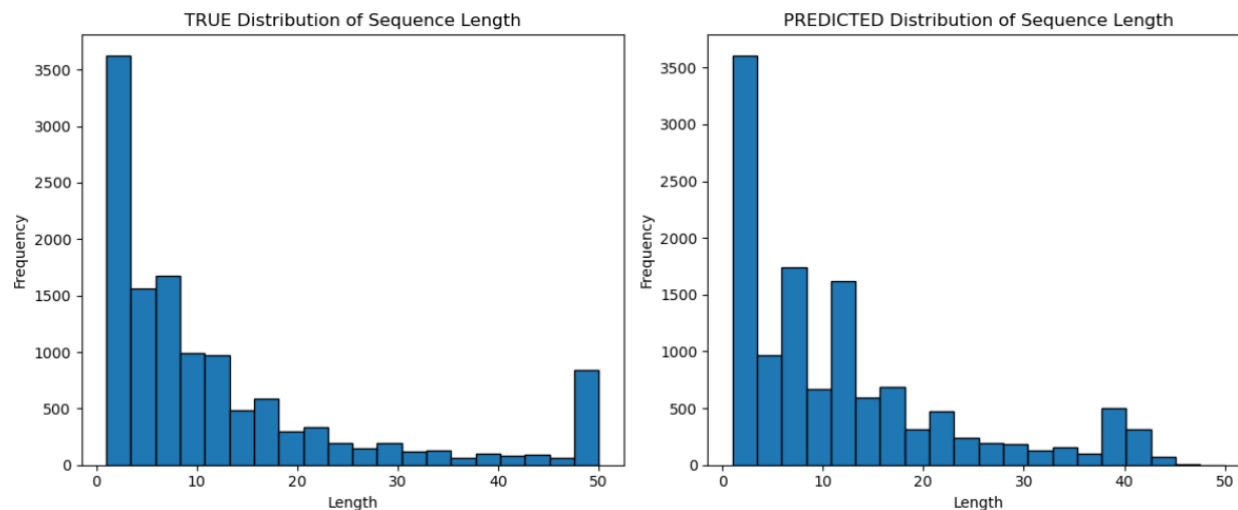


Figure 4-27: Distribution of timestamps of actual and predicted sequence with tuned LSTM using selected features, gender, time_interval_t0 and number of blood draw - Correntropy Loss

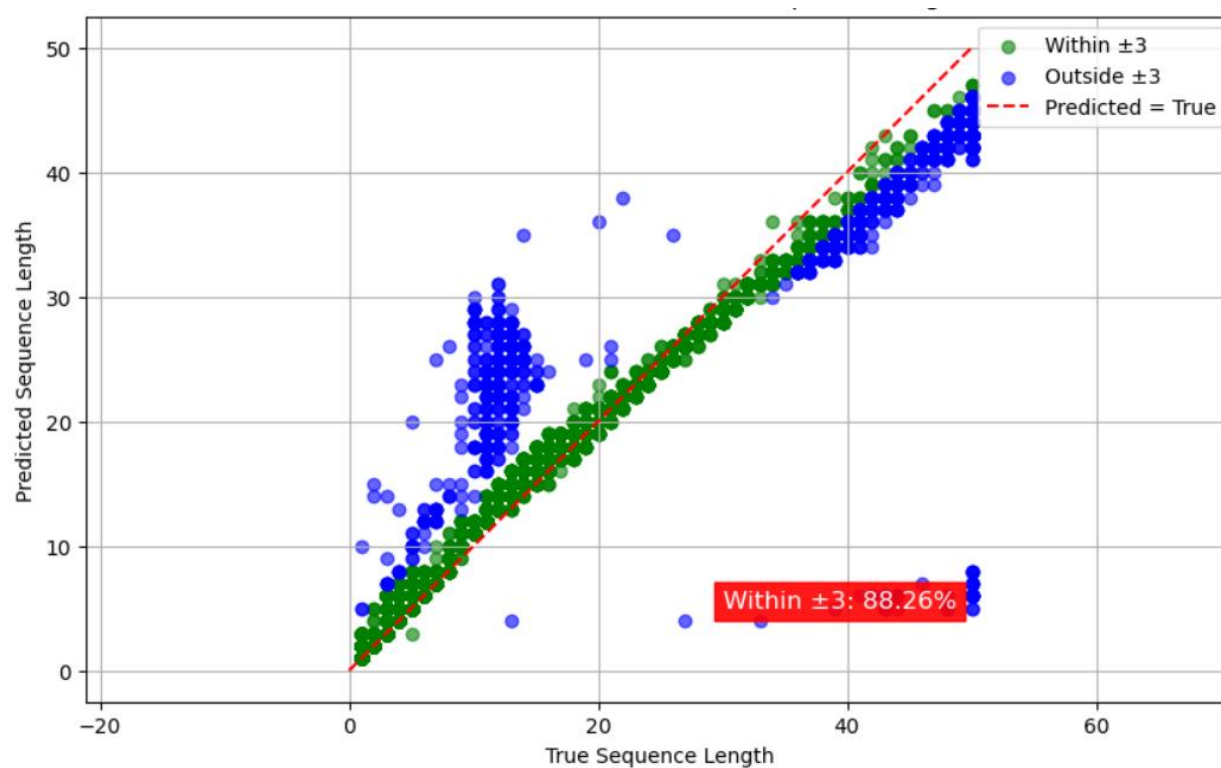


Figure 4-28: Scatter plot of the true vs. predicted sequence length with tuned LSTM using selected features, gender, time_interval_t0 and number of blood draw - Correntropy Loss

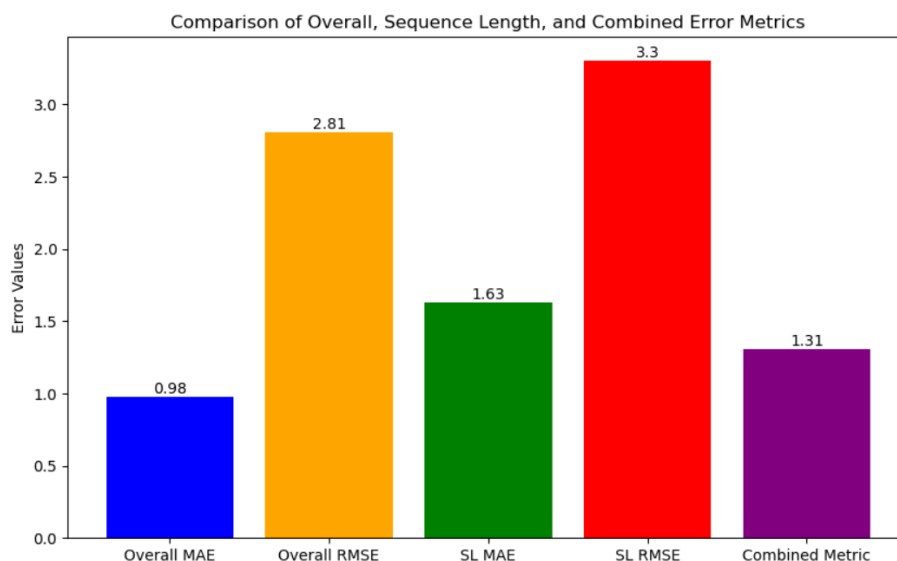


Figure 4-29: Evaluation metrics with tuned LSTM using selected features, gender, time_interval_t0 and number of blood - Correntropy Loss

Table 4-5 provides a summary of the performance metrics, highlighting that the addition of the new feature significantly improved the model's ability to predict sequence length, achieving nearly 89% accuracy across the samples. The selected features, which include additional temporal and quantitative information, such as the time of the first blood draw and the total number of blood draws, along with gender, represent a crucial set of inputs to consider when optimizing the task of predicting blood test frequency for each patient upon ICU admission.

Table 4-5: Comparison of MAE and RMSE for values and sequence lengths across models

Models	Features	Total MAE (hours)	Total MSE (hours)	SL- MAE (unit length)	SL- RMSE (unit length)	R ²	Combined MAE and SL-MAE	Performance in sequence length percentage ±3 blood tests ±MAE in values

Baseline MultiOutput Linear Regression	Removed least significant features and most correlated ones	1.31	3.53	11.54	18.43	-0.05	6.43	45.85%
Simple LSTM with 256 neurons	All features	1.23	3.61	8.5	14.73	-0.07	4.86	54.44%
Simple LSTM with 256 neurons	Removed least significant features and most correlated ones	1.23	3.63	8.73	15.1	-0.06	4.98	54.95%
Simple LSTM with 256 neurons	Adding gender	1.24	3.63	8.77	15.14	-0.06	5.0	55.08%
Simple LSTM with 256 neurons	Adding calcium	1.23	3.62	8.6	14.87	-0.06	4.92	54.87%
Simple LSTM with 256 neurons	Adding chloride	1.23	3.62	8.6	14.87	-0.06	4.92	54.87%
Simple LSTM with 256 neurons	Adding hematocrit	1.23	3.62	8.67	15.0	-0.06	4.95	54.85%
Simple LSTM with 256 neurons (with correntropy Loss – sigma = 1)	Removed least significant features and most	1.23	3.62	8.69	15.03	-0.06	4.96	55.28%

	correlated ones							
Final LSTM model (from hyperparameter tuning, with correntropy Loss – sigma = 1)	Removed least significant features and most correlated ones	0.98	2.81	1.63	3.3	0.35	1.31	88.26%

CHAPTER 5 DISCUSSION

This section presents findings related to the research objective of predicting blood test frequencies using artificial intelligence. It provides an in-depth evaluation of the key features, the loss function, and the metrics used to train an LSTM model. Additionally, it addresses the implications of the results for both practice and theory, acknowledges any limitations, and suggests areas for future research. The section concludes with a summary that reinforces the significance of our findings and demonstrates how this work contributes to the broader academic context.

5.1 Overview of the findings

The primary contribution of this thesis is addressing the gap in blood test management within ICU settings. It changes the focus to forecasting blood test frequencies in advance, as this approach can enhance resource allocation, minimize unnecessary testing, and guide preservation techniques for consecutive tests, ultimately helping to reduce iatrogenic anemia. This predictive capability not only aids healthcare professionals but ultimately enhances patient outcomes. This thesis marks the first application of Long Short-Term Memory (LSTM) networks on medical time series data to predict variable timestamps for blood test frequency. By leveraging a comprehensive medical database with precise monitoring technologies and rigorous protocols from 2012 to 2019, key findings include effective set of features, and the identification of an optimal loss function. These results highlight the potential of artificial intelligence in improving blood test management in critical care settings.

5.1.1 Evaluation of Key Features

First, to evaluate the most selective features, we analyzed their correlation and utilized an ensemble technique to assess their relationship with the target output. For this specific problem, a combination of vital signs, blood test results, demographics, and additional temporal and quantitative features, such as the timing of the first blood draw and the total number of blood draw per patient, proved effective. In the demographic data, adding information like marital status, insurance, or race did not contribute significantly and instead introduced noise, with poor generalization on the test set. As highlighted in other studies, gender and age were identified as

strong demographic indicators. Ultimately, the most effective features were the timing of the first blood draw and the total number of blood tests, because they offer more relevant insights into shorter versus longer sequences. While patient medical information was important for determining specific values, the previous features effectively captured the variability in sequence length. Incorporating the total number of blood draws and the timing of the first blood test into our regression model provided several benefits. Firstly, these features captured crucial information that is likely directly related to the target variable. The R^2 value suggests that incorporating these variables allowed the model to better account for the variability in the data. Additionally, those two features may have interacted with other features, revealing more nuanced data patterns, such as how variations in patient's data changes in relation to other factors. By summarizing key characteristics of the sequences, we provided additional insights into the data, resulting in more robust predictions. This enhancement was confirmed in the final step of our iterative process aimed at improving model performance. For future work, incorporating more informative features can strengthen the model's ability to learn complex relationships and thus using a simple model for multi-task. We can further investigate which factors at the time of ICU admission might impact the number of blood samples taken per patient and present an approximation. Furthermore, factors related to the timing of events, such as the time of day, as well as integrating control-flow information with case data (including physiological inputs, vital signs, and blood results), could prove particularly beneficial [45].

Calcium, chloride, and hematocrit were excluded from the analysis due to their high correlation with other features and increased missing values, indicating redundancy that could affect model efficiency. The most significant predictors, following temporal and quantitative variables, included vital signs such as heart rate, respiratory rate, and systolic blood pressure. Additionally, laboratory results, specifically phosphate, blood urea nitrogen, white blood cell count (WBC), red cell distribution width (RDW), and hemoglobin enhanced model performance, allowing for better training and more accurate inferences. Similar findings have been reported in studies predicting blood test results [11]. It is worth noting that features like the primary diagnosis and the initial care unit were also critical in improving model performance and ensuring effective frequency predictions.

Secondly, our objective was not only to accurately predict the correct interval for each blood draw but also to precisely determine the sequence length. However, achieving accurate sequence length with a single model posed significant challenges, particularly due to the LSTM model's frequent failure to detect the End-of-Sequence token (EOS) effectively. To address this issue, we implemented a pragmatic heuristic approach for cases where accurate detection was not achieved. However, there is a concern that this post-processing might have led to overfitting, necessitating further investigation to ensure generalization to unseen data. We believe this problem is closely related to the complexities of sequence modeling and the inherent limitations of LSTM architectures. One potential solution is to decompose the problem into two sequential tasks: first predicting the total number of blood draws and then using this prediction to inform a second model that forecasts the timings of each blood draw. While this approach could enhance the performance, it would also increase computational complexity and require more extensive pre-processing and post-processing steps. This trade-off between accuracy and efficiency is a crucial consideration in future model designs. Additionally, exploring alternative architectures, such as attention mechanisms or transformer models, could provide more robust handling of variable sequence length and improve EOS token detection. Experimenting with different modeling strategies and incorporating additional contextual features may further enhance our predictions and mitigate the challenges we're currently facing. Ultimately, a thorough evaluation of these strategies will help us achieve a more reliable and effective predictive model.

5.1.2 MSE vs Correntropy Loss

In our exploration of loss functions, we discovered that Mean Squared Error (MSE) was particularly effective for predicting interval timing at each timestamp, whereas Correntropy Loss demonstrated superior performance in capturing variable sequences. This distinction suggests a nuanced approach for future work: we could train an initial model using Correntropy Loss specifically to predict sequence length, and then develop a second model focused on predicting values using MSE. Alternatively, we might consider creating a combined loss function that integrates both MSE and Correntropy Loss. This would allow the model to prioritize specific types of errors, tailoring its performance to the unique challenges of our dataset. As a preliminary

baseline, these findings lay a strong foundation for further analysis. Additional research is essential to refine our methodology and enhance model performance.

5.2 Limitations

This research was confined to patients included in the MIMIC-IV database, which, like many machine learning studies, represents only a fraction of the diverse patient population encountered in the ICU. While a separate test set was created using the same database to assess model performance, the study did not fully account for generalization. This limitation raises concerns about overfitting, highlighting the need for further exploration of the model's performance on unseen data.

In our in-depth analysis, we employed various pre-processing techniques, some of which were beyond the scope of our primary research objectives. While these techniques ultimately influenced model performance, we had to bypass certain methods in favor of simpler approaches. For instance, we addressed missing values by using a boolean indicator alongside the median, and we utilized a fixed word embedding size of 100 without tuning it—a choice that represents a limitation of our work. Additionally, we opted for a hybrid approach to manage outliers instead of thoroughly exploring the most effective methods. This reliance on simpler techniques provided a clearer view of the problem and helped identify a suitable set of features. In fact, imputing missing values with the median can introduce noise, particularly if the missingness is not completely random. Systematic biases in missing data can lead to imputed values that do not accurately reflect the true underlying distribution, potentially undermining the effectiveness of the boolean feature. Further research could explore a more sophisticated way for treating missing values.

Another significant limitation of our study was the strong assumption made regarding the truncation of the target to a maximum of 50 timestamps. This decision, along with reducing blood draws to intervals of less than 24 hours and removing duplicates due to human errors, was intended to simplify the problem initially. However, these assumptions may have inadvertently restricted the complexity and depth of our analysis. By limiting the target sequence length, we may have overlooked important patterns that could emerge over longer timeframes, potentially impacting the model's predictive capabilities.

Additionally, the decision to focus exclusively on patients aged 18 years and older further narrows the applicability of our findings. This age restriction may exclude critical insights from younger populations who may exhibit different blood draw behaviors and medical needs. As a result, the generalizability of our model could be compromised, as it may not adequately represent the full spectrum of patients encountered in clinical settings.

Together, these assumptions not only simplify the problem but also limit the comprehensiveness of our research, potentially affecting the validity and applicability of our findings to real-world scenarios. Future work should consider a more inclusive approach that accounts for a broader range of patients and longer time intervals to enhance the robustness of predictive models in clinical applications.

5.3 Suggestions for Future Research

Predicting multiple regression targets from a single timestamp is a complex task that requires meticulous feature engineering and model tuning. While the research provides valuable insights into the application of LSTMs for this task, it also opens several avenues for further inquiry, particularly in refining feature selection and addressing data variability. The features selected were primarily based on domain knowledge and prior research on blood test results. The LSTM model faced challenges due to data variability, suggesting that the features predominantly represented biomarkers rather than being grounded in a time-series context, which presents an intriguing direction for future exploration.

Also, in future works, integrating this architecture with models that predict blood test results presents an exciting opportunity to enhance clinical decision-making. By developing a unified framework, we could enable dynamic predictions of both optimal blood draw intervals and the expected outcomes of those tests. This integration would allow clinicians to receive timely recommendations based on real-time patient data, improving the overall quality of care. Specifically, the model could indicate not only when to perform blood draws but also forecast what results are likely, providing a comprehensive overview of a patient's condition. This dual capability would facilitate informed decision-making, optimize resource utilization, and ultimately lead to improved patient outcomes reducing iatrogenic anemia.

Moreover, conducting thorough evaluations on diverse patient populations would be crucial to validate the model's effectiveness and generalizability. Incorporating feedback loops from clinicians could also provide valuable insights into the model's predictions, facilitating iterative improvements. Ultimately, exploring these avenues could lead to a powerful tool that enhances clinical workflows and supports personalized patient care in the ICU setting and specifically improves blood management practices.

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

To conclude, our findings directly address significant challenges faced by healthcare professionals in the ICU by providing a framework for optimizing blood management practices. By predicting blood test frequencies upon admission to an ICU ward, clinicians can make more informed decisions about when to draw blood, minimizing unnecessary procedures. This personalized patient-specific approach helps reduce the burden of frequent blood draws, which can lead to patient discomfort and iatrogenic anemia, a common complication resulting from excessive blood draws. Additionally, by offering insights into optimal blood sampling timings, we enable healthcare professionals to manage their resources more effectively, enhancing overall workflow and patient care in the high-pressure ICU environment. Although few studies have utilized machine learning techniques to predict blood test outcomes at specific points in time, this research is the first of its kind that shifts the focus to predicting blood test frequencies upon ICU admission.

In our study, we specifically targeted ICU patients (>18 years old) and utilized key variables for our predictions, including demographic information, temporal and quantitative data, systematically recorded vital signs, and blood test results. These data were sourced from the MIMIC-IV open-source medical research database, which adheres to strict regulatory standards in the ICU setting and functions similarly to an EHR, but with a focus on research. To ensure the reliability and validity of our data, we implemented several key strategies: we conducted thorough checks for completeness, and consistency to address any missing or erroneous entries. Additionally, we employed cross-validation techniques during model training to ensure our findings were robust and not overly reliant on any single subset of data.

Finally, this study focused on LSTM models because prior research has shown their superior performance over many other machine learning algorithms in similar healthcare applications, making them an ideal choice for our work. Moreover, we believe that this approach can be integrated with other studies predicting blood test results, facilitating a multi-tasking framework. By employing Correntropy Loss during training, we tailored the model to our specific needs and improved performance through feature selection techniques, rather than relying on manual testing of combinations. This foundational knowledge can pave the way for future research.

REFERENCES

- [1] P. Olver, M. K. Bohn, and K. Adeli, “Central role of laboratory medicine in public health and patient care,” *Clin. Chem. Lab. Med. CCLM*, vol. 61, no. 4, pp. 666–673, Mar. 2023, doi: 10.1515/cclm-2022-1075.
- [2] R. Juárez-Vela *et al.*, “Related Factors of Anemia in Critically Ill Patients: A Prospective Multicenter Study,” *J. Clin. Med.*, vol. 11, no. 4, p. 1031, Feb. 2022, doi: 10.3390/jcm11041031.
- [3] N. S. Whitehead *et al.*, “Interventions to prevent iatrogenic anemia: a Laboratory Medicine Best Practices systematic review,” *Crit. Care*, vol. 23, no. 1, p. 278, Dec. 2019, doi: 10.1186/s13054-019-2511-9.
- [4] A. J. Ullman, S. Keogh, F. Coyer, D. A. Long, K. New, and C. M. Rickard, “‘True Blood’ The Critical Care Story: An audit of blood sampling practice across three adult, paediatric and neonatal intensive care settings,” *Aust. Crit. Care*, vol. 29, no. 2, pp. 90–95, May 2016, doi: 10.1016/j.aucc.2015.06.002.
- [5] V. Neef *et al.*, “Effect of using smaller blood volume tubes and closed blood collection devices on total blood loss in patients undergoing major cardiac and vascular surgery,” *Can. J. Anesth. Can. Anesth.*, Jan. 2024, doi: 10.1007/s12630-023-02643-8.
- [6] L. J. Matzek *et al.*, “A Contemporary Analysis of Phlebotomy and Iatrogenic Anemia Development Throughout Hospitalization in Critically Ill Adults,” *Anesth. Analg.*, vol. 135, no. 3, pp. 501–510, Sep. 2022, doi: 10.1213/ANE.0000000000006127.
- [7] T. François *et al.*, “Strategies to Reduce Diagnostic Blood Loss and Anemia in Hospitalized Patients: A Scoping Review,” *Pediatr. Crit. Care Med.*, vol. 24, no. 1, pp. e44–e53, Jan. 2023, doi: 10.1097/PCC.0000000000003094.
- [8] C. Delgado-Corcoran, S. Bodily, D. U. Frank, M. K. Witte, R. Castillo, and S. L. Bratton, “Reducing Blood Testing in Pediatric Patients After Heart Surgery: A Quality Improvement Project*,” *Pediatr. Crit. Care Med.*, vol. 15, no. 8, pp. 756–761, Oct. 2014, doi: 10.1097/PCC.0000000000000194.
- [9] L. Devis *et al.*, “Interventions to improve appropriateness of laboratory testing in the intensive care unit: a narrative review,” *Ann. Intensive Care*, vol. 14, no. 1, p. 9, Jan. 2024, doi: 10.1186/s13613-024-01244-y.
- [10] S. Lennox and S. Bench, “Blood sampling in adult critical care: a mixed methods study,” *Int. J. Orthop. Trauma Nurs.*, vol. 45, p. 100923, May 2022, doi: 10.1016/j.ijotn.2022.100923.
- [11] L. Yu, L. Li, E. Bernstam, and X. Jiang, “A deep learning solution to recommend laboratory reduction strategies in ICU,” *Int. J. Med. Inf.*, vol. 144, p. 104282, Dec. 2020, doi: 10.1016/j.ijmedinf.2020.104282.
- [12] T. Huang, L. T. Li, E. V. Bernstam, and X. Jiang, “Confidence-based laboratory test reduction recommendation algorithm,” *BMC Med. Inform. Decis. Mak.*, vol. 23, no. 1, p. 93, May 2023, doi: 10.1186/s12911-023-02187-3.

- [13] L. Yu, Q. Zhang, E. V. Bernstam, and X. Jiang, "Predict or draw blood: An integrated method to reduce lab tests," *J. Biomed. Inform.*, vol. 104, p. 103394, Apr. 2020, doi: 10.1016/j.jbi.2020.103394.
- [14] J. Lee and D. M. Maslove, "Using information theory to identify redundancy in common laboratory tests in the intensive care unit," *BMC Med. Inform. Decis. Mak.*, vol. 15, no. 1, p. 59, Dec. 2015, doi: 10.1186/s12911-015-0187-x.
- [15] A. Ngo, P. Gandhi, and W. G. Miller, "Frequency that Laboratory Tests Influence Medical Decisions," *J. Appl. Lab. Med.*, vol. 1, no. 4, pp. 410–414, Jan. 2017, doi: 10.1373/jalm.2016.021634.
- [16] H. Hjelmgren, B.-M. Ygge, B. Nordlund, and N. Andersson, "Nurses' experiences of blood sample collection from children: a qualitative study from Swedish paediatric hospital care," *BMC Nurs.*, vol. 21, no. 1, p. 62, Dec. 2022, doi: 10.1186/s12912-022-00840-2.
- [17] A. Faisal *et al.*, "Reducing the number of unnecessary routine laboratory tests through education of internal medicine residents," *Postgrad. Med. J.*, vol. 94, no. 1118, pp. 716–719, Dec. 2018, doi: 10.1136/postgradmedj-2018-135784.
- [18] Y. Wu and J. Feng, "Development and Application of Artificial Neural Network," *Wirel. Pers. Commun.*, vol. 102, no. 2, pp. 1645–1656, Sep. 2018, doi: 10.1007/s11277-017-5224-x.
- [19] Q. Zhang *et al.*, "A machine learning framework for identifying influenza pneumonia from bacterial pneumonia for medical decision making," *J. Comput. Sci.*, vol. 65, p. 101871, Nov. 2022, doi: 10.1016/j.jocs.2022.101871.
- [20] B. Mueller, T. Kinoshita, A. Peebles, M. A. Graber, and S. Lee, "Artificial intelligence and machine learning in emergency medicine: a narrative review," *Acute Med. Surg.*, vol. 9, no. 1, p. e740, Jan. 2022, doi: 10.1002/ams2.740.
- [21] J. Gao, "Prediction of sepsis mortality in ICU patients using machine learning methods," 2024.
- [22] Z. Yang, X. Cui, and Z. Song, "Predicting sepsis onset in ICU using machine learning models: a systematic review and meta-analysis," *BMC Infect. Dis.*, vol. 23, no. 1, p. 635, Sep. 2023, doi: 10.1186/s12879-023-08614-0.
- [23] S. Gokhale, "Hospital length of stay prediction tools for all hospital admissions and general medicine populations: systematic review and meta-analysis".
- [24] Md. M. Alam, "An efficient random forest algorithm-based telemonitoring framework to predict mortality and length of stay of patients in ICU," *Multimed. Tools Appl.*, vol. 83, no. 17, pp. 50581–50600, Nov. 2023, doi: 10.1007/s11042-023-17239-z.
- [25] S. L. Hyland, "Early prediction of circulatory failure in the intensive care unit using machine learning," *Nat. Med.*, vol. 26, 2020.
- [26] F. Cismondi *et al.*, "Predicting laboratory testing in intensive care using fuzzy and neural modeling," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, Taipei, Taiwan: IEEE, Jun. 2011, pp. 2096–2103. doi: 10.1109/FUZZY.2011.6007547.

- [27] C. E. Valderrama, D. J. Niven, H. T. Stelfox, and J. Lee, "Predicting Abnormal Laboratory Blood Test Results in the Intensive Care Unit Using Novel Features Based on Information Theory and Historical Conditional Probability: Observational Study," *JMIR Med. Inform.*, vol. 10, no. 6, p. e35250, Jun. 2022, doi: 10.2196/35250.
- [28] S. Xu *et al.*, "Prevalence and Predictability of Low-Yield Inpatient Laboratory Diagnostic Tests," *JAMA Netw. Open*, vol. 2, no. 9, p. e1910967, Sep. 2019, doi: 10.1001/jamanetworkopen.2019.10967.
- [29] G. K. Mahani and M.-R. Pajooan, "Predicting lab values for gastrointestinal bleeding patients in the intensive care unit: A comparative study on the impact of comorbidities and medications," *Artif. Intell. Med.*, vol. 94, pp. 79–87, Mar. 2019, doi: 10.1016/j.artmed.2019.01.004.
- [30] Y. Zhou, F. Tu, K. Sha, J. Ding, and H. Chen, "A Survey on Data Quality Dimensions and Tools for Machine Learning," Jun. 27, 2024, *arXiv*: arXiv:2406.19614. Accessed: Sep. 20, 2024. [Online]. Available: <http://arxiv.org/abs/2406.19614>
- [31] J. Yang *et al.*, "Brief introduction of medical database and data mining technology in big data era," *J. Evid.-Based Med.*, vol. 13, no. 1, pp. 57–69, Feb. 2020, doi: 10.1111/jebm.12373.
- [32] A. E. W. Johnson *et al.*, "MIMIC-IV, a freely accessible electronic health record dataset," *Sci. Data*, vol. 10, no. 1, p. 1, Jan. 2023, doi: 10.1038/s41597-022-01899-x.
- [33] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Min. Knowl. Discov.*, vol. 5, no. 5, pp. 216–233, Sep. 2015, doi: 10.1002/widm.1157.
- [34] S. Hochreiter, Long short-term memory, Neural Computation MIT-Press (1997)
- [35] B.-L. Ye, M. Zhang, L. Li, C. Liu, and W. Wu, "A Survey of Traffic Flow Prediction Methods Based on Long Short-Term Memory Networks," *IEEE Intell. Transp. Syst. Mag.*, pp. 2–27, 2024, doi: 10.1109/MITS.2024.3400679.
- [36] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022, doi: 10.1109/TKDE.2021.3070203.
- [37] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on Multi-Output Learning," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2019, doi: 10.1109/TNNLS.2019.2945133.
- [38] E. Korneva and H. Blockeel, "Towards Better Evaluation of Multi-target Regression Models," in *ECML PKDD 2020 Workshops*, vol. 1323, I. Koprinska, M. Kamp, A. Appice, C. Loglisci, L. Antonie, A. Zimmermann, R. Guidotti, Ö. Özgöbek, R. P. Ribeiro, R. Gavaldà, J. Gama, L. Adilova, Y. Krishnamurthy, P. M. Ferreira, D. Malerba, I. Medeiros, M. Ceci, G. Manco, E. Masciari, Z. W. Ras, P. Christen, E. Ntoutsi, E. Schubert, A. Zimek, A. Monreale, P. Biecek, S. Rinzivillo, B. Kille, A. Lommatzsch, and J. A. Gulla, Eds., in Communications in Computer and Information Science, vol. 1323. , Cham: Springer International Publishing, 2020, pp. 353–362. doi: 10.1007/978-3-030-65965-3_23.

- [39] L. Cai, M. Lei, S. Zhang, Y. Yu, T. Zhou, and J. Qin, “A noise-immune LSTM network for short-term traffic flow forecasting,” *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 30, no. 2, p. 023135, Feb. 2020, doi: 10.1063/1.5120502.
- [40] F. Cismondi *et al.*, “Reducing unnecessary lab testing in the ICU with artificial intelligence,” *Int. J. Med. Inf.*, vol. 82, no. 5, pp. 345–358, May 2013, doi: 10.1016/j.ijmedinf.2012.11.017.
- [41] R. Rios *et al.*, “Handling missing values in machine learning to predict patient-specific risk of adverse cardiac events: Insights from REFINE SPECT registry,” *Comput. Biol. Med.*, vol. 145, p. 105449, Jun. 2022, doi: 10.1016/j.compbimed.2022.105449.
- [42] C. Chen, Y. Tsai, F. Chang, and W. Lin, “Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results,” *Expert Syst.*, vol. 37, no. 5, p. e12553, Oct. 2020, doi: 10.1111/exsy.12553.
- [43] D. Rojo, N. N. Htun, and K. Verbert, “GaCoVi: a Correlation Visualization to Support Interpretability-Aware Feature Selection,” 2020.
- [44] S. Rose, S. Nickolas, and S. Sangeetha, “A recursive ensemble-based feature selection for multi-output models to discover patterns among the soil nutrients,” *Chemom. Intell. Lab. Syst.*, vol. 208, p. 104221, Jan. 2021, doi: 10.1016/j.chemolab.2020.104221.
- [45] M. Camargo, M. Dumas, and O. González-Rojas, “Learning Accurate LSTM Models of Business Processes,” in *Business Process Management*, vol. 11675, T. Hildebrandt, B. F. Van Dongen, M. Röglinger, and J. Mendling, Eds., in Lecture Notes in Computer Science, vol. 11675. , Cham: Springer International Publishing, 2019, pp. 286–302. doi: 10.1007/978-3-030-26619-6_19.

APPENDICES

APPENDIX A ADDITIONAL TABLES

Table A. 1 : Detailed information of the input features detailing the type and units for each

Category	Features	Type	Units	Identifier - table in MIMIC-IV
Demographics	Gender (used in [11], [13])	Categorical binary (nominal)	Male Female	gender - patients
	Age (used in [11], [13])	Integer (discrete) > 18	Year (min 18 years, max 91 years (all patients >91 are all grouped to 91))	anchorage - patients
	Insurance	Categorical (nominal)	MedicareMedicade Other None	insurance - admissions
	Marital status	Categorical (nominal)	'MARRIED' 'SINGLE' 'DIVORCED' 'WIDOWED' Unknown	marital_status - admissions
	Race	Categorical (nominal)	white hispanic/latino - puerto rican multiple race/ethnicity asian - chinese unable to obtain unknown hispanic/latino - salvadoran black/african american white - other european black/caribbean island asian - south east asian black/cape verdean asian black/african white - eastern european other patient declined to answer hispanic or latino hispanic/latino - honduran hispanic/latino - dominican south american portuguese white - russian asian - asian indian hispanic/latino - cuban	race - admissions

			white - brazilian native hawaiian or other pacific islander american indian/alaska native hispanic/latino - guatemalan hispanic/latino - mexican asian - korean hispanic/latino - Colombian hispanic/latino - central american	
Vital signs	Weight	Numerical (continuous)	Kilograms (kg)	patientweight - inputevents
	Heart Rate (used in [11], [13], [29], [40])	Numerical (continuous)	Beat per minute (bpm)	220045 - d_items
	Respiratory Rate (used in [11], [13], [29], [40])	Numerical (continuous)	Inspirations per minute (insp/min)	220210 - d_items
	Oxygen Saturation pulse spo2 (used in [11], [13], [29], [40])	Numerical (continuous)	Percentage (%)	220277 - d_items
	Non invasive Mean systolic blood pressure – Bpsystolic (used in [11], [13])	Numerical (continuous)	Millimeters of mercury (mmHg)	220179 - d_items
	Non invasive Mean diastolic blood pressure – Bpdiaastolic (used in [11], [13])	Numerical (continuous)	Millimeters of mercury (mmHg)	220180 - d_items
	Non invasive Mean blood pressure – Bpmean (used in [11], [13], [29], [40])	Numerical (continuous)	Millimeters of mercury (mmHg)	220181 - d_items

	Temperature (based on TempSite) (used in [40])	Numerical (continuous)	Fahrenheit (°F)	223761 - d_items
Laboratory tests	Sodium - Na (used in [11], [13])	Numerical (continuous)	Milliequivalents per liter (mEq/L)	50983 - d_labitems
	Potassium – K (used in [11], [13])	Numerical (continuous)	Milliequivalents per liter (mEq/L)	50971 - d_labitems
	Chloride – Cl (used in [11], [13])	Numerical (continuous)	Milliequivalents per liter (mEq/L)	50902 - d_labitems
	Bicarbonate – HCO ₃ (used in [11], [13])	Numerical (continuous)	Milliequivalents per liter mEq/L	50882 - d_labitems
	Calcium – Ca (used in [11], [13], [29], [40])	Numerical (continuous)	Milligrams per deciliter (mg/dL)	50893 - d_labitems
	Magnesium – Mg (used in [11], [13])	Numerical (continuous)	Milligrams per deciliter (mg/dL)	50960 - d_labitems
	Phosphate – PO ₄ (used in [11], [13])	Numerical (continuous)	Milligrams per deciliter (mg/dL)	50970 - d_labitems
	Creatinine – Cr (used in [11], [13])	Numerical (continuous)	Milligrams per deciliter (mg/dL)	50912 - d_labitems
	Blood Urea Nitrogen – BUN (used in [11], [13])	Numerical (continuous)	Milligrams per deciliter (mg/dL)	51006 - d_labitems
	Hematocrit (used in [29], [40])	Numerical (continuous)	Percentage (%)	51221 - d_labitems
	Platelet Count – Plt (used in [11], [13], [40])	Numerical (continuous)	Thousands per microliter (K/uL)	51265 - d_labitems
	Hemoglobin – hgb (used in [11], [13], [40])	Numerical (continuous)	Grams per deciliter (g/dL)	51222 - d_labitems

	White Blood Cells (used in [11], [13])	Numerical (continuous)	Thousands per microliter (K/uL)	51301 d_labitems	-
	Mean Corpuscular Hemoglobin Concentration - MCHC	Numerical (continuous)	Percentage (%)	51249 d_labitems	-
	Red Blood Cells - RBC	Numerical (continuous)	Millions per microliter (m/uL)	51279 d_labitems	-
	Mean Corpuscular Volume - MCV	Numerical (continuous)	Femtoliter (fL)	51250 d_labitems	-
	Mean Corpuscular Hemoglobin - MCH	Numerical (continuous)	Picogram (pg)	51248 d_labitems	-
	Red Cell Distribution Width - RDW	Numerical (continuous)	Percentage (%)	51277 d_labitems	-
Clinical Assessment	Primary Diagnosis	Text – version 9 – version 10	ICD codes	icd_code diagnoses_icd (seq_num = 1)	-
	First Care Unit	Categorical (nominal)	Trauma SICU (TSICU), Medical Intensive Care Unit (MICU), Coronary Care Unit (CCU), Cardiac Vascular Intensive Care Unit (CVICU), Medical/Surgical Intensive Care Unit (MICU/SICU)	first_careunit icustays	-

Table A. 2: Results from Grid Search for optimal architecture determination

LSTM Parameter	MAE	MSE	SL MAE	SL MSE	Percentage
{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 2, 'sigma': 1.0}	1.2232	3.6218	8.6747	15.0106	55.1806
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 3, 'sigma': 1.0}	1.2274	3.6216	8.6709	14.9918	55.1488

{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 2, 'sigma': 1.0}	1.2251	3.6179	8.5774	14.8573	55.1329
{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 3, 'sigma': 1.0}	1.2252	3.6224	8.6514	14.9671	55.1249
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 2, 'sigma': 1.0}	1.2245	3.6239	8.6700	14.9855	55.1170
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 4, 'sigma': 1.0}	1.2266	3.6190	8.5669	14.7823	55.0613
{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 2, 'sigma': 1.0}	1.2273	3.6176	8.5983	14.8841	55.0294
{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 3, 'sigma': 1.0}	1.2274	3.6214	8.6600	14.9835	55.0215
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 2, 'sigma': 1.0}	1.2196	3.6148	8.5147	14.6868	55.0215
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 3, 'sigma': 1.0}	1.2243	3.6218	8.7192	15.0687	54.9817
{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 3, 'sigma': 1.0}	1.2247	3.6200	8.6458	14.9626	54.9578
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 2, 'sigma': 1.0}	1.2266	3.6207	8.6389	14.9135	54.9499
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 4, 'sigma': 1.0}	1.2310	3.6330	8.8240	15.2367	54.9339
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 2, 'sigma': 1.0}	1.2252	3.6310	8.7817	15.1639	54.8862
{'batch_size': 64, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 3, 'sigma': 1.0}	1.2283	3.6301	8.7558	15.1352	54.8623
{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 2, 'sigma': 1.0}	1.2263	3.6216	8.6576	14.9742	54.8544

{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 3, 'sigma': 1.0}	1.2283	3.6318	8.7271	15.0749	54.8544
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 3, 'sigma': 1.0}	1.2299	3.6328	8.8304	15.2586	54.8146
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 4, 'sigma': 1.0}	1.2258	3.6301	8.7334	15.0754	54.7430
{'batch_size': 64, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 4, 'sigma': 1.0}	1.2265	3.6221	8.6719	14.9768	54.7350
{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 512, 'num_layers': 4, 'sigma': 1.0}	1.2255	3.6285	8.6899	15.0075	54.6395
{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 512, 'num_layers': 4, 'sigma': 1.0}	1.2293	3.6284	8.7965	15.1860	54.6315
{'batch_size': 512, 'dropout_rate': 0.2, 'initial_units': 256, 'num_layers': 4, 'sigma': 1.0}	1.2331	3.6413	8.8127	15.2007	54.5679
{'batch_size': 512, 'dropout_rate': 0.5, 'initial_units': 256, 'num_layers': 4, 'sigma': 1.0}	1.2292	3.6292	8.7159	15.0564	54.4803

APPENDIX B ADDITIONAL FIGURES

```

WITH unique_samples AS
( SELECT DISTINCT icu.stay_id, lab.charttime, lab.specimen_id
  FROM      `physionet-data.mimiciv_icu.icustays` AS icu
  JOIN      `physionet-data.mimiciv_hosp.labevents` AS lab
  ON        icu.hadm_id = lab.hadm_id
  JOIN      `physionet-data.mimiciv_hosp.d_labitems` AS dlab_item
  ON        lab.itemid = dlab_item.itemid
  WHERE     dlab_item.fluid = 'Blood'
  AND       lab.charttime >= icu.intime
  AND       lab.charttime <= icu.outtime
  AND       icu.los < 50 ),
distinct_samples AS
( SELECT      stay_id, charttime
  FROM        unique_samples
  GROUP BY    stay_id, charttime )
SELECT  stay_id, COUNT(*) AS blood_sample_count
FROM    distinct_samples
GROUP BY stay_id
ORDER BY stay_id;

```

Figure B.1: SQL query to validate the total number of blood draws per patient during ICU stay from admission to discharge

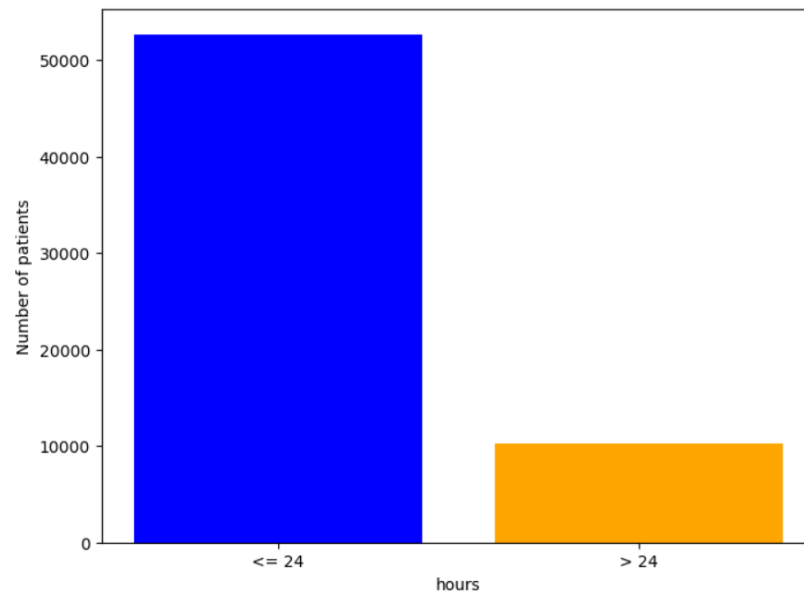


Figure B.2: Distribution of patients undergoing blood draws at 24-Hour intervals or longer

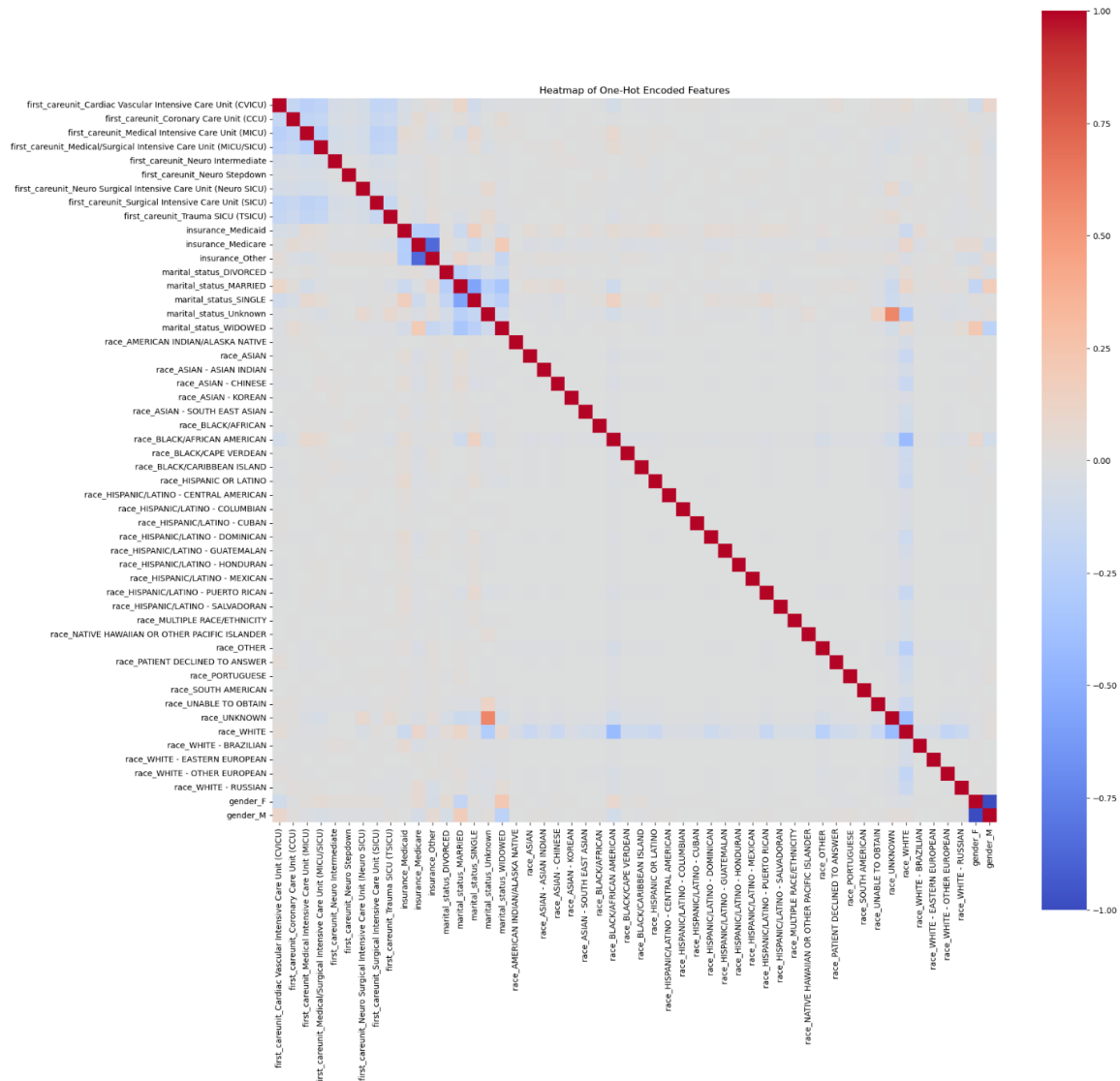
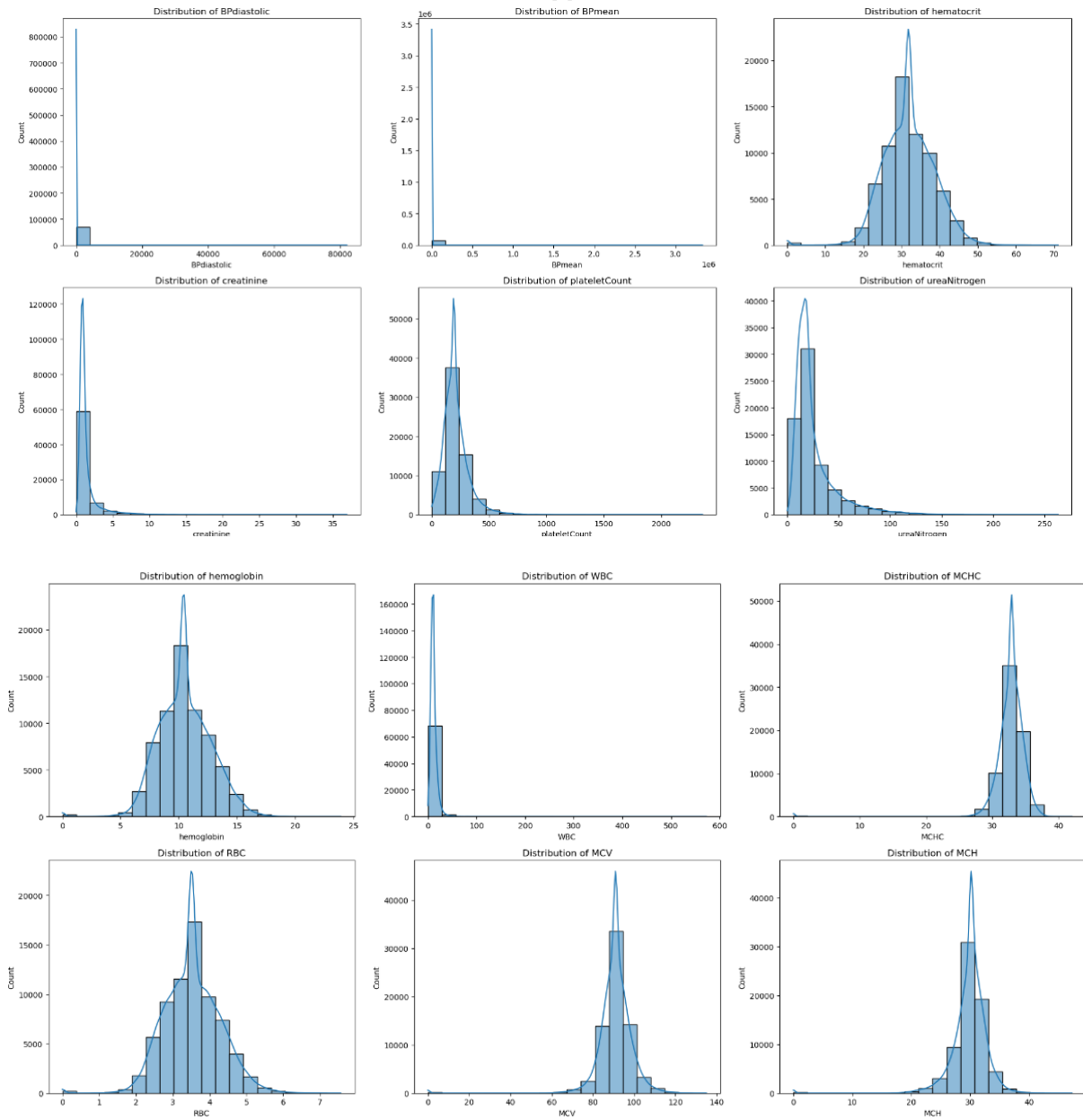
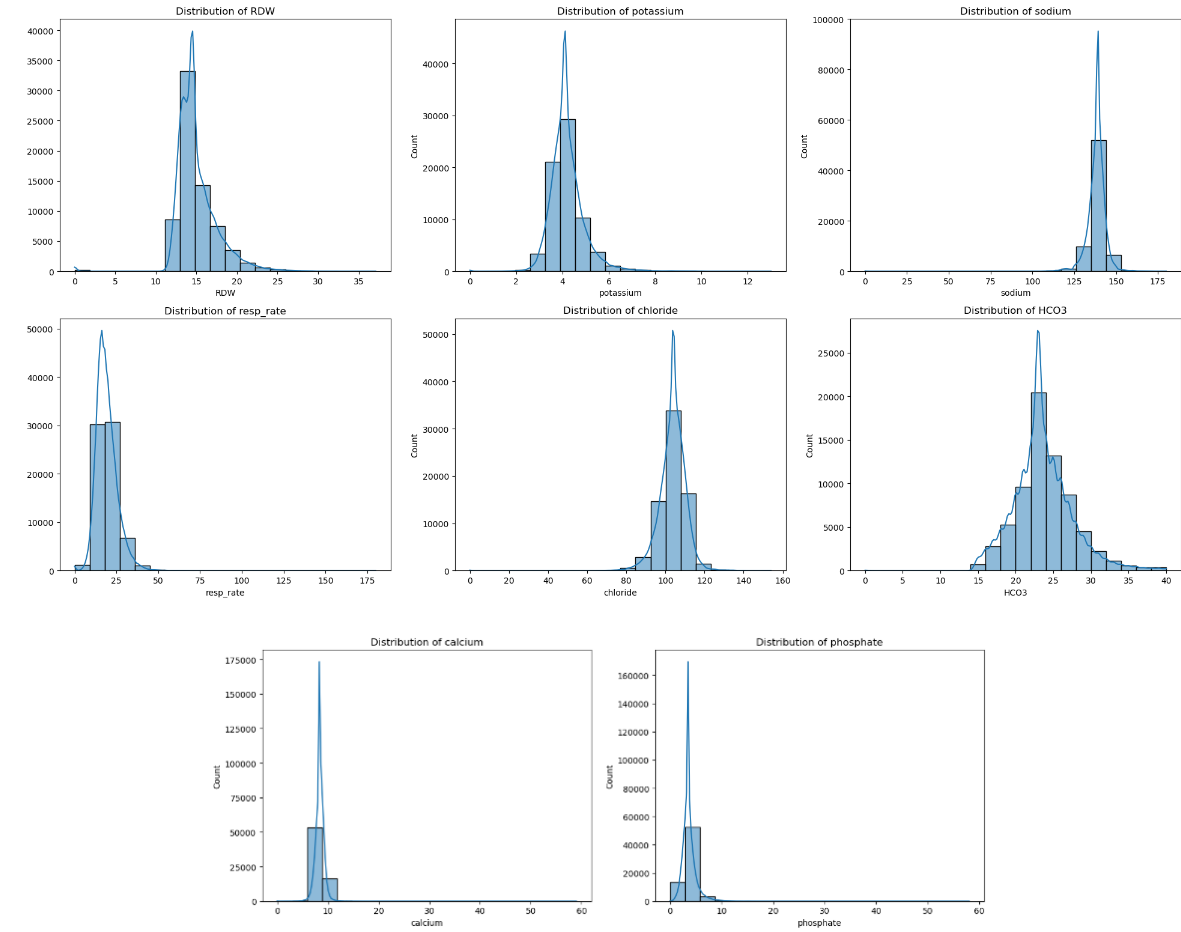
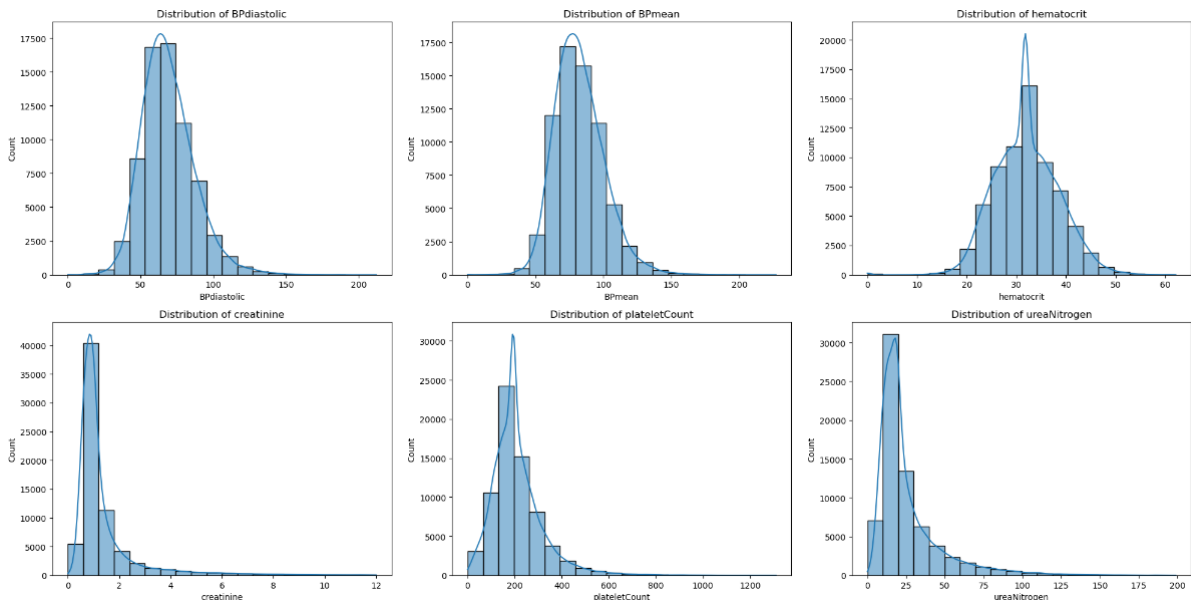


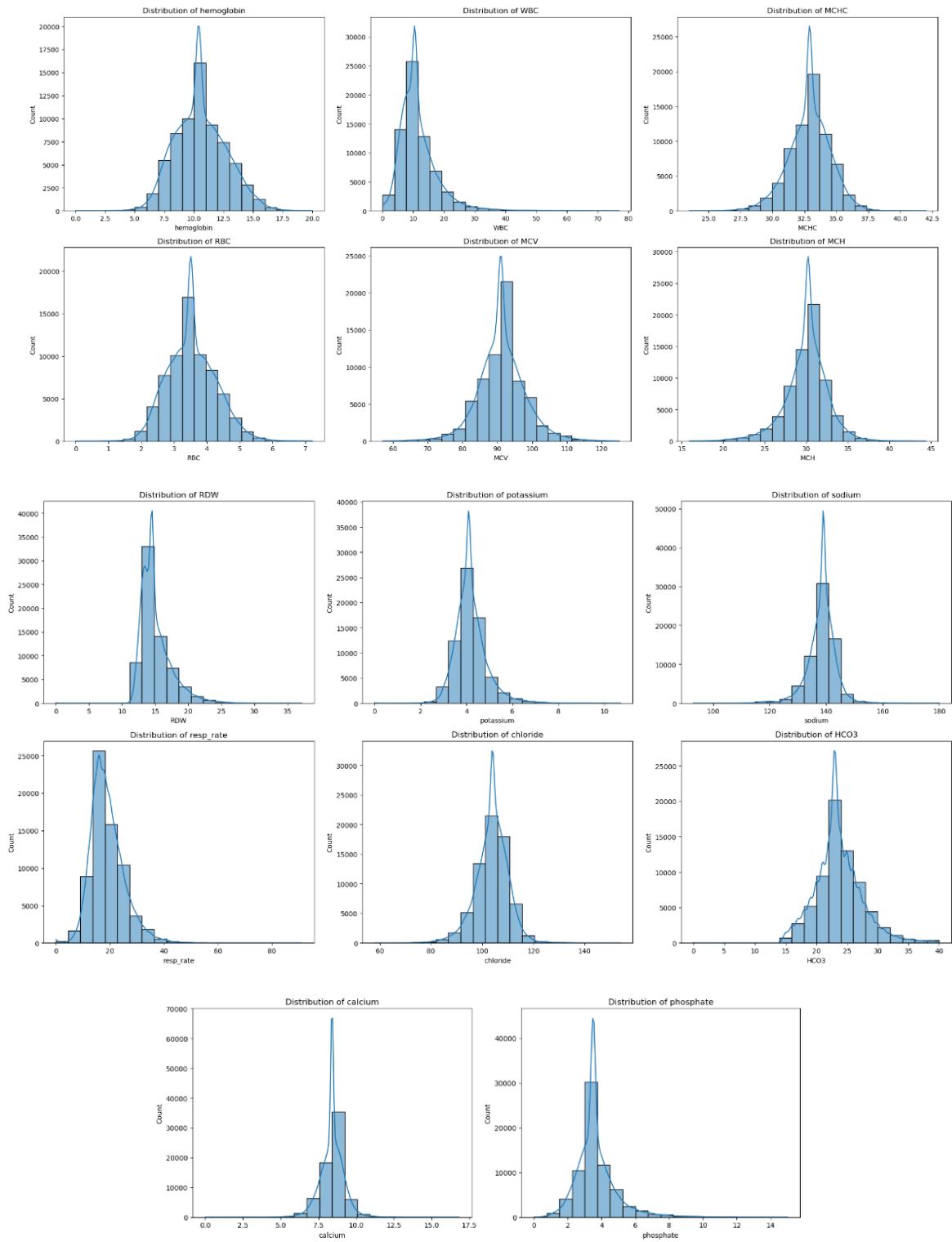
Figure B.3: Correlation heatmap for categorical features including for demographics and clinical characteristics





a)





b)

Figure B.4: Distribution of numerical features – a) before applying outlier removal – b) after applying outlier removal techniques using hybrid of Z-score and IQR

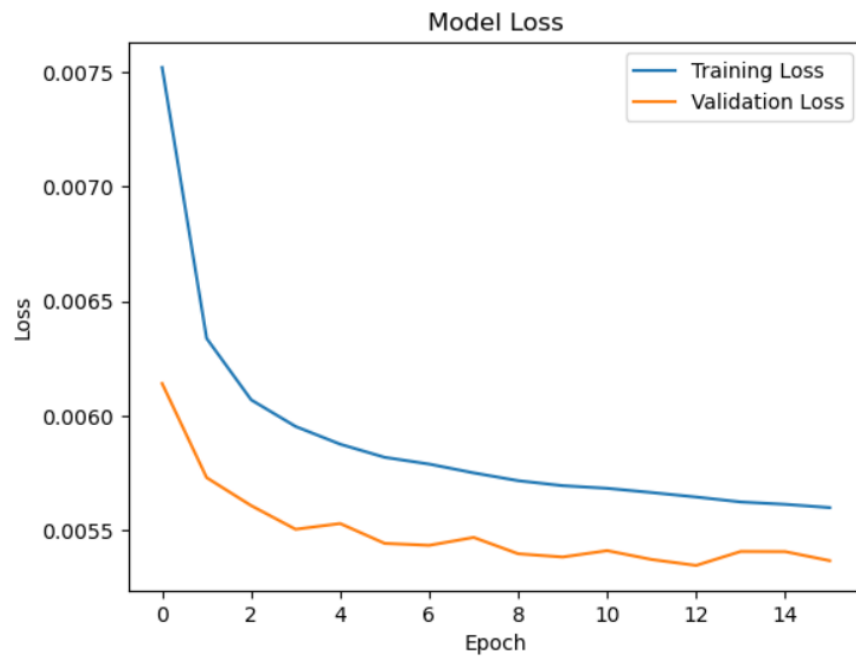


Figure B.5: Training the final LSTM model using cross-validation with 20% and early stopping with patience=3

APPENDIX C PYTHON CODE

C. 1 - Removing Outliers in numerical features

```
In [26]: def remove_outliers(df, features, z_threshold=3, iqr_multiplier=1.5):
    cleaned_data = df.copy()
    outlier_counts = {}

    for feature in features:
        original_count = cleaned_data.shape[0]
        skew = cleaned_data[feature].skew()

        if abs(skew) < 0.5: # Approximately normal distribution
            # Calculate z-scores and remove outliers
            z_scores = np.abs((cleaned_data[feature] - cleaned_data[feature].mean()) / cleaned_data[feature].std())
            cleaned_data = cleaned_data[z_scores < z_threshold]
            print(feature)
        else: # Significantly skewed distribution
            # Calculate IQR and remove outliers
            Q1 = cleaned_data[feature].quantile(0.25)
            Q3 = cleaned_data[feature].quantile(0.75)
            IQR = Q3 - Q1
            if (feature == 'creatinine'):
                cleaned_data = cleaned_data[~((cleaned_data[feature] < 0) | (cleaned_data[feature] > 12))]
            else:
                cleaned_data = cleaned_data[~((cleaned_data[feature] < (Q1 - iqr_multiplier * IQR)) | (cleaned_data[feature]
                > (Q3 + iqr_multiplier * IQR)))]

        # Count the number of outliers removed
        removed_count = original_count - cleaned_data.shape[0]
        outlier_counts[feature] = removed_count

    return cleaned_data, outlier_counts

# Remove outliers with adjusted thresholds
df_cleaned, outlier_counts = remove_outliers(df, numerical_features, z_threshold=5, iqr_multiplier=10.0)

print("Original DataFrame shape:", df.shape)
print("Cleaned DataFrame shape:", df_cleaned.shape)
print("Outliers removed from each feature:", outlier_counts)
```

C. 2- Word embedding for primary diagnosis

```
In [40]: from gensim.models import Word2Vec
import pandas as pd
import re

data = pd.read_csv('diagnoses.csv')

# Function to preprocess text - and tokenize
def preprocess_text(text):
    # Remove punctuation marks except for hyphens and apostrophes
    text = re.sub(r'[^\w\s\-\']', '', text)
    tokens = text.lower().split()
    return tokens

# Tokenize and preprocess textual descriptions
tokenized_descriptions = [preprocess_text(text) for text in data['long_title']]

# Train Word2Vec model on tokenized descriptions
model = Word2Vec(sentences=tokenized_descriptions, vector_size=100, window=5, min_count=1, sg=1)

# Function to get embedding vector for a diagnosis
def get_embedding(diagnosis):
    try:
        tokens = preprocess_text(diagnosis)
        vector_sum = sum(model.wv[token] for token in tokens if token in model.wv.key_to_index)
        return vector_sum / len(tokens) # Average vector for tokens in the diagnosis
    except KeyError:
        return [0] * model.vector_size # Handle unknown diagnoses by returning zeros

# Preprocess diagnoses in X to match the preprocessing of long_title
X['diagnoses_processed'] = X['diagnoses'].apply(preprocess_text)

# Apply embedding to X
X['diagnoses_word2vec'] = X['diagnoses_processed'].apply(
    lambda tokens: sum(model.wv[token] for token in tokens if token in model.wv.key_to_index) / len(tokens))
```

C. 3 - Simple LSTM model

```
In [116]: time_steps = 1
          features = len(X_train.columns.tolist())
          max_output_length = 50

          # Create a simple LSTM model with no additional Hidden Layers
          model = Sequential()
          model.add(LSTM(256, activation='relu', input_shape=(time_steps, features))) # Input Layer
          model.add(RepeatVector(max_output_length)) # Repeat the output for each timestamp
          model.add(LSTM(max_output_length, activation='relu', return_sequences=True)) # Output Layer
          model.add(TimeDistributed(Dense(1))) # Outputs one value for each timestamp

          # Compile the model
          model.compile(optimizer='adam', loss='mse')

          model.summary()

          # Reshape the input data to (samples, time_steps, features)
          X_train_resaped = X_train.values.reshape((X_train.shape[0], time_steps, features))
          X_test_resaped = X_test.values.reshape((X_test.shape[0], time_steps, features))

          # Reshape y_train and y_test to have the shape (samples, max_output_length, 1)
          y_train_resaped = y_train.reshape((y_train.shape[0], max_output_length, 1))
          y_test_resaped = y_test.reshape((y_test.shape[0], max_output_length, 1))

          # Set up early stopping
          early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)

          # Train the model with a validation split of 20%
          model.fit(X_train_resaped, y_train_resaped,
                    epochs=50,
                    batch_size=64,
                    validation_split=0.2, # Use 20% of the training data for validation
                    callbacks=[early_stopping],
                    verbose=1)
```

C. 4 - Final LSTM with best performance

```
In [130]: time_steps = 1
          features = len(X_train.columns.tolist())
          max_output_length = 50

          # Create the LSTM model
          model = Sequential()
          model.add(LSTM(512, return_sequences=True, activation='relu', input_shape=(time_steps, features)))
          model.add(Dropout(0.5))
          model.add(LSTM(256, activation='relu', return_sequences=True))
          model.add(Dropout(0.5))
          model.add(LSTM(128, activation='relu', return_sequences=False))
          model.add(Dropout(0.5))
          model.add(RepeatVector(max_output_length))
          model.add(LSTM(max_output_length, activation='relu', return_sequences=True))
          model.add(TimeDistributed(Dense(1))) # Outputs one value for each timestamp

          # Compile the model with the custom loss function
          model.compile(optimizer='adam', loss=correntropy_loss(sigma=1.0))

          # # Compile the model
          #model.compile(optimizer='adam', loss="mse")

          model.summary()
```