

ICU MORTALITY PREDICTION USING MACHINE LEARNING TECHNIQUES

Submitted in partial fulfilment of the requirements for the award of

Master of Science degree in
Bioinformatics and Data science

By

INFANTA PRINCY J (42739014)



DEPARTMENT OF BIOINFORAMATICS AND DATA SCIENCE
SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL – 2024



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

www.sathyabama.ac.in

DEPARTMENT OF BIOINFORMATICS

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **J INFANTA PRINCY (42739014)**, who carried out the final year project entitled **"ICU MORTALITY PREDICTION USING MACHINE LEARNING TECHNIQUES"** under our supervision from January 2024 to April 2024.

Dr. SWETHA SUNKAR

Internal Guide

Dr. JEMMY CHRISTY

Head of the Department

Submitted for Viva Voce Examination held on **20/04/2024**

Internal Examiner

External Examiner

DECLARATION

I, **J. INFANTA PRINCY**, hereby declare that the Project Report entitled **ICU MORTALITY PREDICTION USING MACHINE LEARNING TECHNIQUES** done by me under the guidance of **Dr. SWETHA SUNKAR** at Sathyabama Institute of Science and Technology, Jeppiaar Nagar, Rajiv Gandhi Road, Chennai-600119, Tamil Nadu, India, is submitted in partial fulfilment of the requirements for the award of **Master of Science in Bioinformatics and Data Science**.

DATE: 24/04/2024



PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENTS

I am delighted to express my heartfelt acknowledgement to the **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement and support throughout the duration of my project. Their guidance has been invaluable in ensuring the successful completion of this project, and I am sincerely grateful for their contribution.

I convey thanks to **DR. REKHA CHAKRAVARTHY**, Dean, School of science and humanities, for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to **Dr. H. JEMMY CHRISTY**, Head of the Bioinformatics Department, for her support has been instrumental in helping me to overcome obstacles and achieve success in this project.

I am deeply indebted to my Project Guide, **DR. SWETHA SUNKAR**, for her invaluable guidance, suggestions, and constant encouragement throughout the project. Her expertise and insights have been instrumental in shaping my work and ensuring its successful completion.

I wish to extend my gratitude to all Teaching and Non-Teaching staff members for their support and assistance in various ways, which have contributed to the successful completion of this project.

J INFANTA PRINCY

ABSTRACT

Accurate prediction of mortality rates in the Intensive Care Unit (ICU) is crucial for optimizing patient care and resource allocation. In this study, the performance of five machine learning algorithms—Random Forest, Decision Tree, XGBoost, Logistic Regression, and Recurrent Neural Network (RNN)—for ICU mortality prediction using the area under the receiver operating characteristic curve (AUROC) as the primary evaluation metric. Leveraging a comprehensive dataset encompassing clinical variables, demographics, and outcomes from ICU patients, a comparative analysis of these algorithms in terms of predictive accuracy, interpretability, and computational efficiency. Our results demonstrate that Random Forest and RNN exhibit superior performance with high AUROC scores, indicating their effectiveness in mortality risk assessment and clinical decision-making. XGBoost offers interpretable results while Decision Tree and Logistic Regression specific strengths and challenges. This study contributes valuable insights into the application of machine learning techniques for ICU mortality prediction, emphasizing the importance of selecting appropriate algorithms based on AUROC performance and clinical requirements.

Keywords: ICU mortality prediction, machine learning, Random Forest, Decision Tree, XGBoost, Logistic Regression, recurrent neural network, AUROC.

TABLE OF CONTENTS

S. NO	TOPICS	PAGE NO
	List of Abbreviations	iv
	List of Tables	v
	List of Figures	v
1.	INTRODUCTION	6
	1.1 Background	6
	1.2 Intensive Care Unit	7
	1.3 Machine Learning in healthcare	7
	1.3.1 Logistic Regression	8
	1.3.2 Random Forest	8
	1.3.3 Decision Tree	9
	1.3.4 Extreme Gradient Boost	9
	1.3.5 Recurrent Neural Network	10
	1.4 Mortality Predictive Model	10
2.	REVIEW OF LITERATURE	11
3.	AIM AND SCOPE	15
	3.1 Aim	15
	3.2 Scope	15
4.	MATERIALS AND METHODS	16
	4.1 Data collection	16
	4.2 Preprocessing	16
	4.3 Data Visualization	16
	4.4 Feature selection	17
	4.5 Model training	17
	4.6 Evaluation	19
	4.7 Software and Tools	19
5.	RESULTS & DISCUSSION	22
	5.1 Data collection	22
	5.2 Preprocessing	23

	5.3 Data Visualization	24
	5.4 Feature Selection	30
	5.5 Model Training	33
	5.6 Evaluation: AUROC	33
6.	SUMMARY AND CONCLUSION	35
	6.1 Summary	35
	6.2 Conclusion	35
7.	REFERENCES	37
8.	APPENDIX	40

LIST OF ABBREVIATIONS

APACHE	Acute Physiology and Chronic Health Evaluation
AST	<i>Aspartate aminotransferase</i>
AUROC	Area Under the Receiver Operating Characteristic curve
BP	Blood Pressure
EDA	Exploratory Data Analysis
HER	Electronic Health Records
GCS	Glasgow Coma Scale
GRU	Gated Recurrent Unit
ICU	Intensive Care Unit
LDH	Lactic Dehydrogenase
LOS	Length of Stay
MIMIC	The Medical Information Mart for Intensive Care ²²
ML	Machine Learning
MPM	Mortality Probability Model
O2 SAT	Oxygen Saturation level
PT	Prothrombin Time
RES_RATE	Respiratory rate
RFE	Recursive Feature Elimination
RNN	Recurrent Neural Network
SAPS	Simplified Acute Physiology Score
TEMP	Temperature
WBC	White Blood Cells
XGBOOST	Extreme Gradient Boost

LIST OF TABLES

TABLES	TITLE	PAGE.NO
5.1.1	Dataset	23
5.6.2	Evaluation Results	34

LIST OF FIGURES

FIGURES	TITLE	PAGE NO
5.1.2	Architecture of the framework of this study	22
5.1.3	The process of the selection	24
5.3.1	Age distribution of ICU admitted patients	25
5.3.2	ICU patient's distribution of Length of Stay in days	26
5.3.3	Gender distribution	27
5.3.4	Mortality Counts	27
5.3.5	Diagnoses	28
5.3.6	Mortality rate for each diagnosis	29
5.4.1	Correlation Heatmap	30
5.4.2	Features	31
5.4.3	Feature Importance	32
5.6.1	AUROC Curve	33

CHAPTER 1

INTRODUCTION

In the high-stakes environment of the Intensive Care Unit (ICU), understanding and predicting mortality rates are paramount for effective patient care. ICU patients often present with complex and life-threatening conditions, ranging from severe infections like sepsis to organ failures and traumatic injuries. Identifying the main reasons for mortality in the ICU helps healthcare providers prioritize interventions, allocate resources efficiently, and tailor treatment plans to maximize patient outcomes.

The primary benefit of mortality prediction in the ICU is its role in resource allocation. By identifying patients with a higher risk of mortality, healthcare teams can ensure that these individuals receive the necessary level of monitoring, specialized interventions, and expert care. This targeted approach not only improves patient survival rates but also optimizes resource utilization within the ICU, ensuring that critical resources such as ventilators, medications, and skilled personnel are allocated where they are most needed.

1.1 BACKGROUND

Mortality risk prediction helps clinicians make better decisions in patient healthcare. However, existing severity scoring systems or algorithms used in intensive care units (ICUs) often rely on the laborious manual collection of complex variables and lack sufficient validation in diverse clinical environments, thus limiting their practical applicability. This study aims to evaluate the performance of machine learning models that utilize routinely collected clinical data for short-term mortality risk prediction (Zhao et al., 2023).

A potentially beneficial source to assist healthcare providers is the vast amount of data captured. However, these huge volumes of medical data such as patient characteristics, medication administration records, and genomic sequences, make it bewildering and perhaps impossible to make decisions as an individual. Fortunately, prediction models powered by Machine Learning (ML) are capable of learning and providing tangible insights and it is no surprise that it is reported that such models are becoming a necessity for modern health systems (Ge et al, 2018).

1.2 INTENSIVE CARE UNIT

The ICU is a critical hospital department with a trained medical crew providing emergency care for life-threatening conditions. During crises like natural hazards and pandemics, providing immediate health services becomes crucial for emergencies.

ICU health records provide a comprehensive overview of the patient's medical history, treatment course, and outcomes during their stay in the intensive care unit, serving as a critical source of information for healthcare providers to deliver optimal care and ensure patient safety. Identification of patients at high risk of death in the ICU is important for guiding treatment decisions, quality assurance and resource utilization management. Several scoring systems have been developed for this purpose, including Acute Physiology and Chronic Health Evaluation (APACHE) such as APACHE III¹ and APACHE IV²; Simplified Acute Physiology Score (SAPS) such as SAPS II³, SAPS III⁴, and Mortality Probability Model (MPM) such as MPM I⁵, MPM II⁶ and MPM III^{7,5}. Recently, there has been an increasing interest in applying more advanced machine learning models to ICU mortality prediction (Ge et al, 2018).

1.3 MACHINE LEARNING IN HEALTHCARE

Machine learning (ML) has emerged as a powerful tool in healthcare for analyzing complex datasets and extracting valuable insights to improve patient outcomes, streamline clinical workflows, and enhance healthcare delivery (Zhao et al., 2023).

The limitations of existing scoring systems have led to the emergence of novel machine learning (ML) algorithms for mortality prediction. It is inspiring that ML algorithms can be continuously trained using newly obtained data in clinical settings, allowing for updates and recalibration over time. This dynamic nature improves the accuracy of the model as populations and treatments evolve. Over the past few decades, researchers have demonstrated that ML techniques outperform traditional prediction methods in predicting outcomes such as mortality, complications, and length of stay (Zhao et al., 2023). In healthcare, machine learning algorithms are applied to a wide range of tasks, including disease diagnosis, prognosis, treatment planning, and patient monitoring. By leveraging patterns and relationships hidden within large volumes of healthcare data, machine learning models can provide predictive analytics, personalized recommendations, and decision support tools for healthcare providers.

Moreover, machine learning techniques enable the integration of diverse data sources, such as electronic health records (EHRs), medical imaging, genomics, and wearable sensors, to generate actionable insights and facilitate precision medicine approaches tailored to individual patient needs.

1.3.1 LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of an event occurring or not occurring. It's particularly useful when the dependent variable (the one you're trying to predict) is categorical with two possible outcomes, such as yes/no, success/failure, or in medical contexts, alive/dead.

In the context of ICU (Intensive Care Unit) mortality prediction, logistic regression can be a valuable tool. Medical professionals often use predictive models to assess the likelihood of patients surviving or dying during their ICU stay. This information helps them make more informed decisions about treatment plans and resource allocation.

1.3.2 RANDOM FOREST

Random Forest is a powerful ensemble learning technique that combines multiple decision trees to make predictions. Each tree in the forest is trained on a random subset of the data and a random subset of features, which helps reduce overfitting and improve generalization.

The RF is a supervised Decision Tree based ML algorithm, which has the capability of coping with the overfitting problem. This ensemble method constructs a multitude of decision trees on different samples and utilizes them for classifying an element based on the majority vote. Alongside making predictions, RF is capable of determining variable importance according to their impact on predicting the target variable (Saadatmand et al., 2022).

1.3.3 DECISION TREE

Decision Trees are fundamental building blocks used in ensemble methods like Random Forests. They recursively partition the feature space based on thresholds that optimize purity or information gain at each node. Decision Trees are easy to

understand and visualize, making them valuable for interpreting the logic behind predictions in ICU mortality prediction models.

However, they can be prone to overfitting, especially with complex datasets. Techniques like pruning and using ensemble methods (like Random Forest) help mitigate this issue. Decision Trees provide a foundation for more advanced models and are often used in combination with other algorithms for improved prediction accuracy.

1.3.4 EXTREME GRADIENT BOOSTING

XGBoost (Extreme Gradient Boosting) is an ensemble learning technique that uses gradient boosting algorithms to build a predictive model by sequentially adding weak learners (typically decision trees) to the ensemble. XGBoost is known for its high predictive accuracy and efficiency, making it well-suited for large-scale datasets commonly found in healthcare settings.

In ICU mortality prediction, XGBoost can handle diverse types of features, automatically handle missing values, and effectively capture nonlinear relationships between predictors and outcomes. Its ability to incorporate regularization techniques helps prevent overfitting and improves generalization performance. XGBoost's interpretability and feature importance analysis also provide valuable insights for clinicians in understanding mortality risk factors.

1.3.5 RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data by retaining memory of previous inputs. In the context of ICU mortality prediction, RNNs can capture temporal dependencies inpatient data, such as trends in vital signs over time.

This makes them suitable for modelling dynamic patient trajectories and predicting outcomes based on evolving clinical conditions. However, training RNNs can be computationally intensive, and they may require a large amount of data to generalize effectively. Preprocessing steps like normalization and feature engineering are crucial for optimizing RNN performance in healthcare applications.

1.4 ICU MORTALITY PREDICTIVE MODEL

Mortality models in Intensive Care Units (ICU) are important for clinical decision-support tasks such as identifying high-risk patients and prioritizing their care.⁶ ICU mortality prediction involves the estimation of the likelihood of a patient dying during their stay in the ICU. Accurate prediction of ICU mortality is crucial for informing clinical decision-making, resource allocation, and end-of-life care planning. Machine learning techniques, including logistic regression and recurrent neural networks, have been widely employed for ICU mortality prediction by leveraging diverse patient data, such as demographic information, clinical variables, laboratory results, and physiological measurements.

By analysing patterns and trends in patient data, predictive models can identify risk factors associated with ICU mortality and provide early warning signs to healthcare providers. The development of accurate ICU mortality prediction models can help prioritize interventions, allocate resources efficiently, and improve patient outcomes in critical care settings.

CHAPTER 2

REVIEW OF LITERATURE

Zhao et al., (2023) Developed a machine learning model using 12,393 ICU patients' routine clinical data. The model used four different algorithms: logistic regression, random forest, XGboost, and artificial neural network. The model improved mortality risk prediction within a 1-week time frame, comparing discrimination and calibration performance. This study evaluates the performance of machine learning models for short-term mortality risk prediction in intensive care units (ICUs). The study used 12,393 ICU patients and four machine learning algorithms: logistic regression, randomforest, extreme gradient boosting (XGboost), and artificial neural network. The XGBoost algorithm demonstrated the highest performance, with an AUROC of 0.9702, AUPRC of 0.8517, and a favorable Brier score of 0.0259 for 24-hour mortality risk prediction. Although performance decreased for larger time windows, it still achieved comparable results for 3-day mortality risk prediction.

Lubinger et al., (2023) analysed large data sets to classify, predict, and gain useful conclusions using AI and Machine learning. Observational multicenter cohort study. All COVID-19 adult patients admitted to 25 ICUs belonging to the VENETO ICU network (February 28th 2020-april 4th 2021) were enrolled. Patients admitted to the ICUs before 4th March 2021 were used for model training while patients admitted after the 5th of March 2021 were used for external validation. A further group of patients, admitted to the ICU of IRCCS Ca' Granda Ospedale Maggiore Policlinico of Milan, was used for external validation. A SuperLearner machine learning algorithm was applied for model development, and both internal and external validation was performed. Clinical variables available for the model were (i) age, gender, sequential organ failure assessment score, Charlson Comorbidity Index score (not adjusted for age), Palliative Performance Score; (ii) need of invasive mechanical ventilation, non-invasive mechanical ventilation, O2 therapy, vasoactive agents, extracorporeal membrane oxygenation, continuous venous-venous hemofiltration, tracheostomy, re-intubation, prone position during ICU stay; and (iii) re-admission in ICU. One thousand two hundred ninety-three (80%) patients were included in the "training set", while 124 (8%) and 199 (12%) patients were included in the "test set 1" and "test set 2," respectively. Three different predictive models were developed. Each model included different sets.

of clinical variables. The three models showed similar predictive performances, with a training balanced accuracy that ranged between 0.72 and 0.90, while the cross-validation performance ranged from 0.75 to 0.85. Age was the leading predictor for all the considered models.

Saadatmand et al., (2023) The COVID-19 pandemic has significantly impacted health systems, particularly Intensive Care Units (ICUs), leading to overburdening staff and resource shortages. This research aims to use Machine Learning (ML) on hospital data to support hospital managers and practitioners in managing COVID-19 patients. The study uses five ML algorithms: eXtreme Gradient Boosting, K-Nearest Neighbor, Random Forest, bagged-CART, and LogitBoost. The ensemble stacking approach, implementing Neural Net or General Linear Model, achieves the best results for ICU admission prediction, with slightly better results for mortality. This approach serves as an example for future pandemics or crises.

Kim et al., (2022) aimed to improve predictive models for ICU inpatients by periodically updating them with the latest clinical data. Improving predictive models for intensive care unit (ICU) inpatients requires a new strategy that periodically includes the latest clinical data and can be updated to reflect local characteristics. The extracted data from all adult patients admitted to the ICUs of two university hospitals with different characteristics from 2006 to 2020, and a total of 85,146 patients were included in this study. Machine learning algorithms were trained to predict in-hospital mortality. The predictive performance of conventional scoring models and machine learning algorithms was assessed by the area under the receiver operating characteristic curve (AUROC). The conventional scoring models had various predictive powers, with the SAPS III (AUROC 0.773 [0.766–0.779] for hospital S) and APACHE III (AUROC 0.803 [0.795–0.810] for hospital G) showing the highest AUROC among them. The best performing machine learning models achieved an AUROC of 0.977 (0.973–0.980) in hospital S and 0.955 (0.950–0.961) in hospital G. The use of ML models in conjunction with conventional scoring systems can provide more useful information for predicting the prognosis of critically ill patients. In this study, the predictive model can be made more robust by training with the individual data of each hospital.

Lorenzoni et al., (2021) A study enrolled COVID-19 adult patients from 25 ICUs in the VENETO ICU network from February 28th 2020 to April 4th 2021. The model was developed using a SuperLearner machine learning algorithm and tested on patients before and after March 5th, 2021. The model included clinical variables such as age, gender, organ failure assessment score, and palliative performance score. Three different predictive models were developed, with similar accuracy and cross-validation performance. Age was the leading predictor for all models. This article was published in Springer Link. The study aimed to develop a machine learning model to predict ICU mortality in COVID-19 patients using predefined clinical parameters. The model was developed using a SuperLearner machine learning algorithm and tested on 25 ICUs in the VENETO ICU network. Three different predictive models were developed, each incorporating different clinical variables. The models showed similar predictive performances, with a training balanced accuracy of 0.72 to 0.90 and cross-validation performance of 0.75 to 0.85. Age was the leading predictor for all models.

Ge et al., (2018) evaluated an interpretable ICU mortality prediction model using Recurrent Neural Networks (RNN) with long short-term memory units. Most existing studies used logistic regression to establish scoring systems to predict intensive care unit (ICU) mortality. Machine learning-based approaches can achieve higher prediction accuracy but, unlike the scoring systems, frequently cannot provide explicit interpretability. They evaluated an interpretable ICU mortality prediction model based on Recurrent Neural Networks (RNN) with long short-term memory units. This model combines both sequential features with multiple values over the patient's hospitalization (e.g. vital signs or laboratory tests) and non-sequential features (e.g. diagnoses), while identifying features that most strongly contribute to the outcome. Using a set of 4,896 MICU admissions from a large medical center, the model achieved a c-statistic for prediction of ICU mortality of 0.7614 compared to 0.7412 for a logistic regression model that used the same data, and identified clinically valid predictors.

Ghanvatkar, Rajan (2019) Mortality models in ICUs are crucial for clinical decision support, identifying high-risk patients and prioritizing care. Previous mortality models have used predictive variables mainly from Electronic Medical Records (EMR) where each patient observation can be represented as a sparse multivariate time series.

Bedside monitors are another common data source in ICUs containing high resolution time series, which have not been explored in combination with EMR data for mortality modelling. It is the first step towards building such a model. Specialized techniques developed for sparse time series cannot be used to model multiple time series at different resolutions. To address this problem, develop MTS-RNN, a new deep recurrent neural network architecture. Our preliminary experiments on real clinical data show that MTS-RNN outperforms state-of-the-art mortality models in predictive accuracy, highlighting the importance of using clinical time series at multiple resolutions for ICU mortality prediction.⁶ This short paper published at AIS Electronic Library (AISeL).

Yan et al., (2020) To support decision making and logistical planning in healthcare systems, this study leverages a database of blood samples from 485 infected patients in the region of Wuhan, China, to identify crucial predictive biomarkers of disease mortality. For this purpose, machine learning tools selected three biomarkers that predict the mortality of individual patients more than 10 days in advance with more than 90% accuracy: lactic dehydrogenase (LDH), lymphocyte and high-sensitivity C- reactive protein (hs-CRP). In particular, relatively high levels of LDH alone seem to play a crucial role in distinguishing the vast majority of cases that require immediate medical attention. This finding is consistent with current medical knowledge that high LDH levels are associated with tissue breakdown occurring in various diseases, including pulmonary disorders such as pneumonia. Overall, this Article suggests a simple and operable decision rule to quickly predict patients at the highest risk, allowing them to be prioritized and potentially reducing the mortality rate.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The aim of this study is to develop and evaluate predictive models for ICU mortality within the first three days of admission using machine learning techniques, such as random forest, decision tree, extreme gradient boost, logistics regression and recurrent neural network (RNNs). Accurate prediction of ICU mortality is crucial for informing clinical decision-making, resource allocation, and end-of-life care planning.

3.2 SCOPE

- To collect and preprocess comprehensive patient datasets containing demographic, clinical, and physiological variables relevant to ICU mortality prediction.
- To implement logistic regression, decision tree, extreme gradient boost, random forest and recurrent neural network models for ICU mortality prediction, utilizing appropriate feature selection and engineering techniques.
- To evaluate the predictive performance of those models using standard evaluation metrics such as accuracy and area under the receiver operating characteristic curve (AUC-ROC).
- To compare the performance of models in terms of predictive accuracy, interpretability, and computational efficiency.

CHAPTER 4

MATERIALS AND METHODS

4.1 DATA COLLECTION

- ❑ A comprehensive dataset from multiple ICU facilities, including clinical variables such as age, gender, comorbidities, vital signs (e.g., heart rate, blood pressure), and laboratory values (e.g., white blood cell count, creatinine levels).
- ❑ The dataset also included outcomes such as ICU mortality status (0 for survival, 1 for mortality) and length of ICU stay.

4.2 PREPROCESSING

- ❑ A data preprocessing steps, including:
 - Handling missing values: Imputed missing data using methods such as mean imputation, median imputation, or predictive imputation based on other features.
 - Scaling numerical features: Normalized numerical features to a common scale (e.g., z-score normalization) to ensure fair comparison among variables.

4.3 DATA VISUALIZATION

Data visualization is a critical component of data analysis and communication, involving the creation of graphical representations that convey complex information in a clear and meaningful way. By transforming raw data into visual formats such as charts, graphs, maps, and dashboards, data visualization enables users to identify patterns, trends, outliers, and relationships within datasets, facilitating data-driven decision-making and insights generation. Effective data visualization not only enhances data comprehension but also improves storytelling, aiding in the communication of findings, recommendations, and actionable insights to stakeholders, clients, and audiences across various domains including business,

academia, healthcare, and government. With the advent of interactive and dynamic visualization tools and libraries like Matplotlib, Seaborn, data visualization has become more accessible, engaging, and impactful, empowering users to explore, analyse, and present data in visually compelling and informative ways.

4.4 FEATURE SELECTION

- A feature selection to identify relevant predictors for ICU mortality prediction. Common techniques included:

- Correlation analysis: Examined correlations between features and removed highly correlated variables to reduce multicollinearity.
- Recursive Feature Elimination (RFE): Used iterative model-based selection to identify the most important features for prediction.

Feature engineering to create new informative features, such as interaction terms, polynomial features, or time-based features

4.5 MODEL TRAINING

- Split the pre-processed dataset into training and testing sets (e.g., 70% for training, 30% for testing) or utilized cross-validation techniques (e.g., k-fold cross-validation) to assess model generalization.
- The following machine learning algorithms for ICU mortality prediction:
 - **Random Forest:** The RF is a supervised Decision Tree based ML algorithm, which has the capability of coping with the overfitting problem (Breiman, 2001). This ensemble method constructs a multitude of decision trees on different samples and utilizes them for classifying an element based on the majority vote (Oshiro et al., 2012). Alongside making predictions, RF is capable of determining variable importance

according to their impact on predicting the target variable (Boulesteix et al., 2012).

- **Decision Tree:** Single decision tree model that partitions data based on feature thresholds to make predictions. decision trees are quite versatile and can handle both categorical and numerical data, making them suitable for analyzing diverse patient variables in ICU settings. They're also easy to visualize and understand, which can be beneficial for clinical interpretation. One advantage of decision trees is their ability to capture complex nonlinear relationships between predictors and outcomes. In the context of ICU mortality prediction, this flexibility allows the model to learn intricate patterns and decision rules from the data.
- **XGBoost (Extreme Gradient Boosting):** Extreme gradient boosting (XGB) is a celebrated ensemble ML algorithm based on the gradient boosted decision trees framework, which works more efficiently in terms of speed and performance compared to most ML approaches (Chen and Guestrin, 2016). Higher values of this parameter make the model more complex and may lead to overfitting. Eta, which lies within 0 and 1, controls the learning rate. Gamma determines the minimum loss reduction for making a split. Column sample by tree specifies the subsample ratio of columns when a new tree is constructed. Minimum child weight is the minimum sum of sample weight required in a child. Subsample denotes the ratio of the training instances.
- **Logistic Regression:** Logistic regression is a commonly used statistical technique for binary classification tasks, such as predicting the likelihood of an event occurring or not occurring. In the context of healthcare and ICU settings, logistic regression can be applied to predict binary outcomes, such as patient survival or mortality. Logistic regression is a powerful tool for predicting binary outcomes in healthcare settings. One advantage of logistic regression is its interpretability. The coefficients of

the logistic regression model can provide insights into the relative importance and direction of influence of each predictor variable on the outcome, aiding clinicians in understanding risk factors and making informed decisions.

- **Recurrent Neural Network (RNN):** RNNs can be highly effective in capturing temporal dependencies and patterns in ICU data, especially when dealing with time-series data such as patient vital signs, lab results, and treatment interventions recorded at regular intervals. This temporal information can be crucial for predicting patient outcomes like mortality. GRUs is a type of RNN can effectively capture temporal dynamics in patient data, such as vital signs, lab results, and treatment trajectories, to predict patient outcomes with improved accuracy and efficiency compared to traditional RNN architectures.

4.6 EVALUATION

- **Model evaluation metrics:** Evaluated model performance using metrics such as accuracy, precision, recall, F1 score, area under the receiver operating characteristic curve (AUC-ROC), and confusion matrix analysis.
- **Comparative analysis:** Compared the performance of each algorithm based on the evaluation metrics to identify the most effective model for ICU mortality prediction.

4.7 SOFTWARE AND TOOLS

- **Anaconda** is a popular open-source distribution platform for Python and R programming languages that is widely used in data science, machine learning, and scientific computing. One of Anaconda's key advantages is its comprehensive package management system, which simplifies the installation and management of libraries, dependencies, and development environments. Anaconda comes pre-packaged with a vast array of data science tools,

including the Python programming language, Jupyter Notebooks, NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, and many others, making it a one-stop solution for data scientists, researchers, and developers.

- **Jupyter Notebook** is an interactive computational environment that supports multiple programming languages like Python, R, and Julia, enabling users to create documents combining live code, visualizations, explanatory text, and mathematical equations. Its cell-based structure allows for iterative code development and immediate feedback, making it ideal for data analysis, machine learning prototyping, and scientific exploration. With seamless integration of popular libraries and the ability to export notebooks in various formats, Jupyter Notebook facilitates reproducible research, collaborative work, and effective communication of findings, making it a go-to tool for data scientists, researchers, educators, and developers.
- **Python** is a versatile and widely used programming language in the field of data science, machine learning, and artificial intelligence. Its simplicity, readability, and extensive library support make it a preferred choice for developers and researchers alike. Python's rich ecosystem of libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn provides powerful tools for data manipulation, analysis, and visualization, making it ideal for building predictive models and conducting data-driven research.
- **Scikit-learn** is a popular machine learning library in Python that offers a comprehensive suite of tools for various machine learning tasks. It provides easy-to-use interfaces for implementing classification, regression, clustering, dimensionality reduction, and model selection techniques. Scikit-learn's well-documented API, efficient algorithms, and built-in model evaluation metrics make it accessible for both beginners and experienced practitioners. Its integration with other Python libraries allows seamless data preprocessing, feature engineering, and model deployment workflows, making it a choice for prototyping and productionizing machine learning solutions.

- **TensorFlow** is a powerful open-source machine learning framework developed by Google that focuses on deep learning and neural network-based models. TensorFlow's core strengths lie in its ability to efficiently handle large-scale datasets, perform distributed computing, and train complex deep learning architectures. It provides a flexible and scalable platform for building neural networks, including recurrent neural networks (RNNs) models. TensorFlow's computational graph abstraction and automatic differentiation capabilities simplify the process of designing, training, and fine-tuning deep learning models, making it a preferred framework for cutting-edge research and production-level deployments in areas like computer vision, natural language processing, and reinforcement learning.

CHAPTER 5

RESULTS AND DISCUSSION

The proposed framework of this study steps is illustrated in Fig. 1. The figure demonstrates the database and the series of actions that are carried out, to prepare the data for modelling. These includes integration of datasets, data cleaning, dealing with missing values, balancing the dataset, and feature selection which will form the basis for the ML models to be applied. The next step applies five different ML algorithms to make prediction and evaluate them based on the statistical index and ROC curves (Saadatmand et al., 2022).

The below flow chart shows the architecture of this thesis:

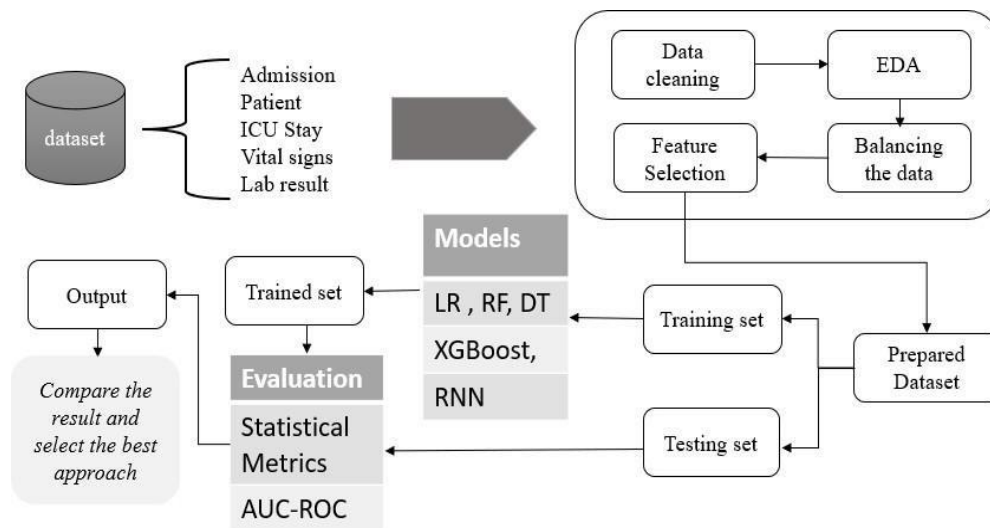


Fig. 5.1.1 Architecture of the framework of this study

All these steps are applied for the five ML models. These processes are described in detail in the following sections. All analyses and modelling are done in Python, which is an open-source programming language.

5.1 DATA COLLECTION

Data collection is a critical step in any data-driven project, including ICU mortality prediction models. The quality, quantity, and relevance of the data collected directly

impact the accuracy and reliability of the predictive models built upon it. In the context of ICU mortality prediction, several key considerations and challenges arise in the process of data collection.

row_id	subject_id	hadm_id	icustay_id	Dbsource	first_careunit	last_careunit	first_wardid	last_wardid
12742	10006	142345	206504	Carevue	MICU	MICU	52	52
12747	10011	105331	232110	Carevue	MICU	MICU	15	15
12749	10013	165520	264446	Carevue	MICU	MICU	15	15
12754	10017	199207	204881	Carevue	CCU	CCU	7	7
12755	10019	177759	228977	Carevue	MICU	MICU	15	15
12762	10026	103770	277021	Carevue	SICU	SICU	33	33
12763	10027	199395	286020	Carevue	CSRU	CSRU	12	12
12765	10029	132349	226055	Carevue	SICU	SICU	33	33
12767	10032	140372	267090	Carevue	MICU	MICU	52	52
12768	10033	157235	254543	Carevue	SICU	SICU	33	33
12770	10035	110244	296804	Carevue	SICU	SICU	23	23
12771	10036	189483	227834	Carevue	MICU	MICU	12	12
12773	10038	111115	235482	Carevue	SICU	SICU	33	33
12775	10040	157839	272047	Carevue	MICU	MICU	12	12
12777	10042	148562	258147	Carevue	CSRU	CSRU	14	14
12778	10043	168674	266122	Carevue	MICU	MICU	12	12
12779	10044	124073	270154	Carevue	MICU	MICU	52	52
12780	10045	126949	203766	Carevue	MICU	SICU	50	57

Table 5.1.2 - dataset

The above Table shows the data which is collection from **MIMIC-III** (The Medical Information Mart for Intensive Care III) database. The dataset contains 136 rows and 53 columns.

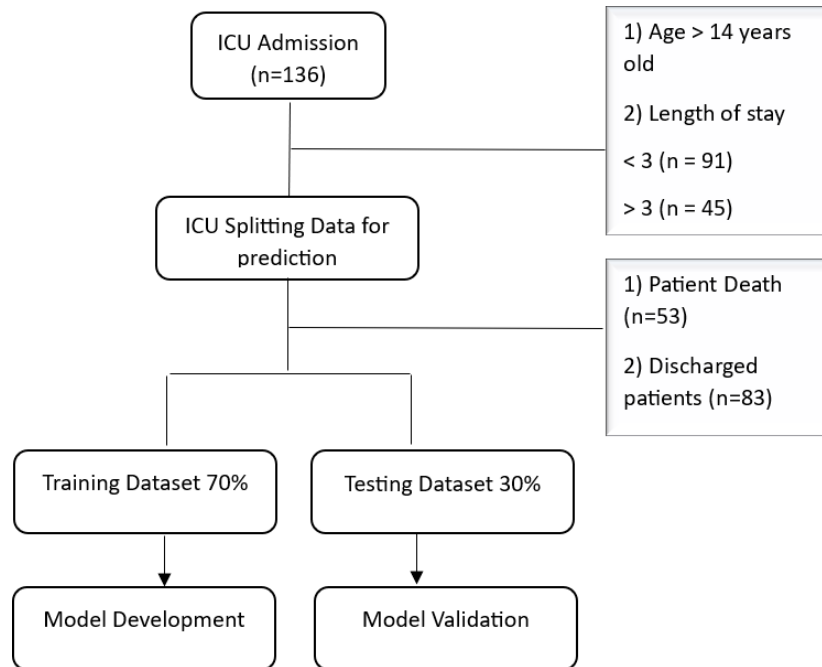


Fig. 2 The process of the selection

5.2 DATA PREPROCESSING

Data preprocessing plays a crucial role in the success of predictive modelling, especially in the context of ICU mortality prediction where the data is often complex, noisy, and heterogeneous. Several key discussions arise when considering data preprocessing steps for this specific application.

One important aspect is handling missing data. In ICU datasets, missing values are common due to various reasons such as sensor malfunctions, incomplete records, or patient-specific factors. Imputation techniques such as mean imputation can be used to fill in missing values, but the choice of imputation method can impact model performance and should be carefully considered. Additionally, the extent of missing data and its potential impact on predictive accuracy need to be evaluated to ensure robustness in the preprocessing stage.

5.3 DATA VISUALIZATION

Data visualization techniques such as scatter plots, histograms, and box plots are essential for EDA. They help identify patterns, trends, outliers, and distributional characteristics in ICU mortality prediction data. Visualization of variables like vital signs, lab results, and demographic factors can reveal potential correlations and insights into mortality risk factors.

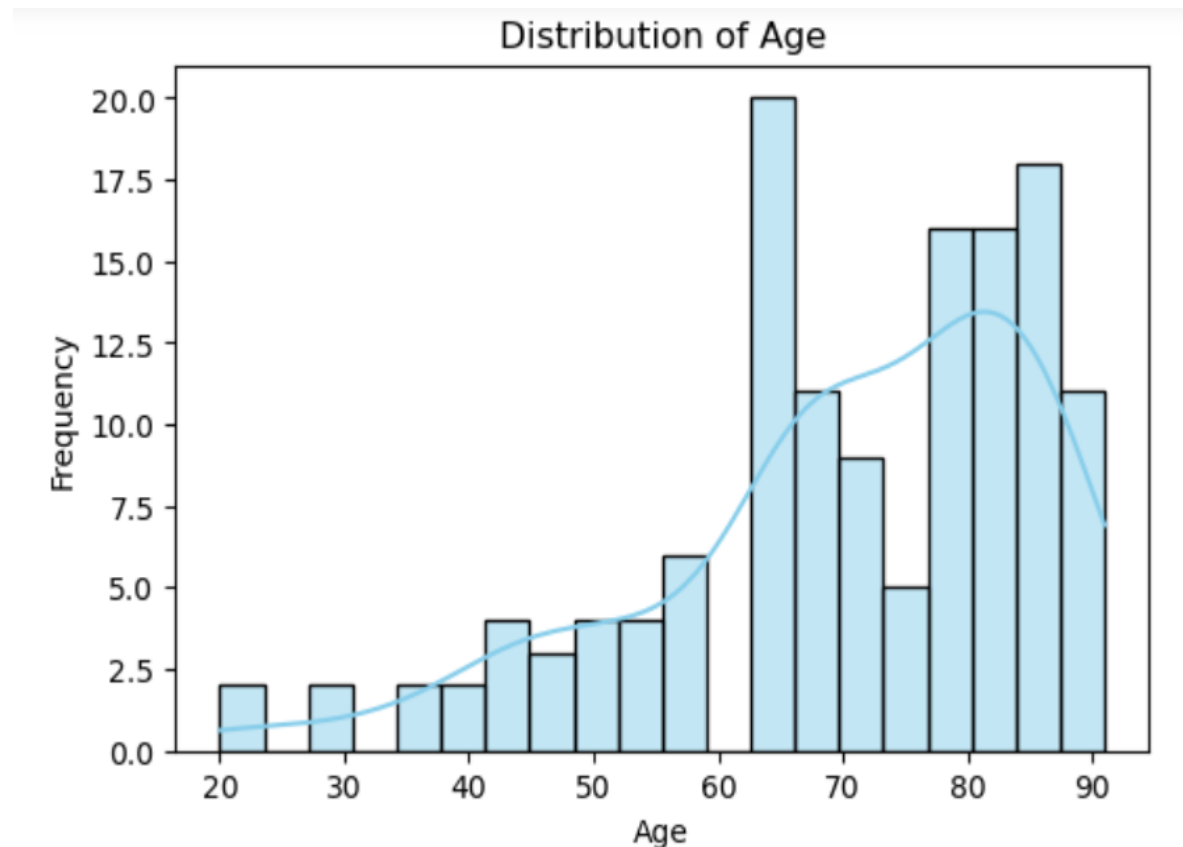


Fig.5.3.1 - Age distribution of ICU admitted patients

Age is often considered a significant factor in predicting mortality among ICU patients. The distribution of age within this context can vary widely, the fig.5.3.1 shows that older age groups (above 60) have higher mortality rates.

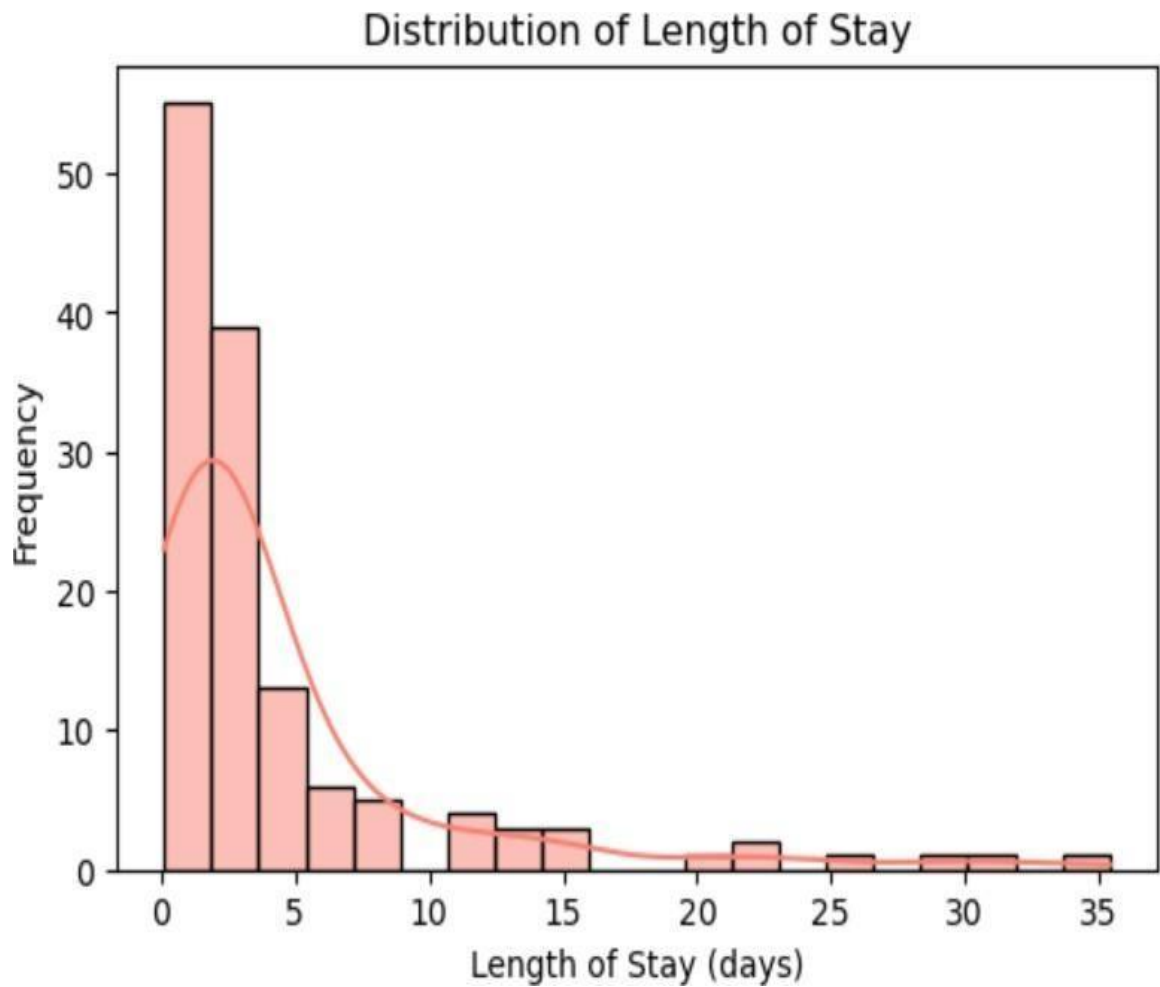


Fig.5.3.2 - ICU patient's distribution of Length of Stay in days

The above a bar graph of LOS can show us the distribution of stays, whether it's skewed towards shorter or longer durations, which can inform feature selection and modelling decisions.

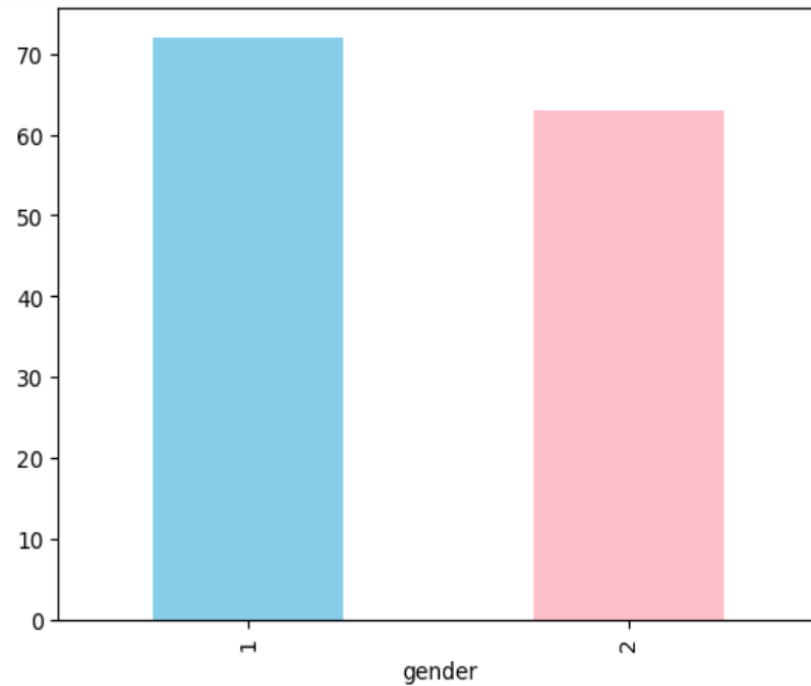


Fig.5.3.3 - Gender distribution

Gender is another factor that often plays a role in ICU mortality prediction. The gender distribution fig.5.3.3 – 1 represents male count (70) and 2 represents female count (60).

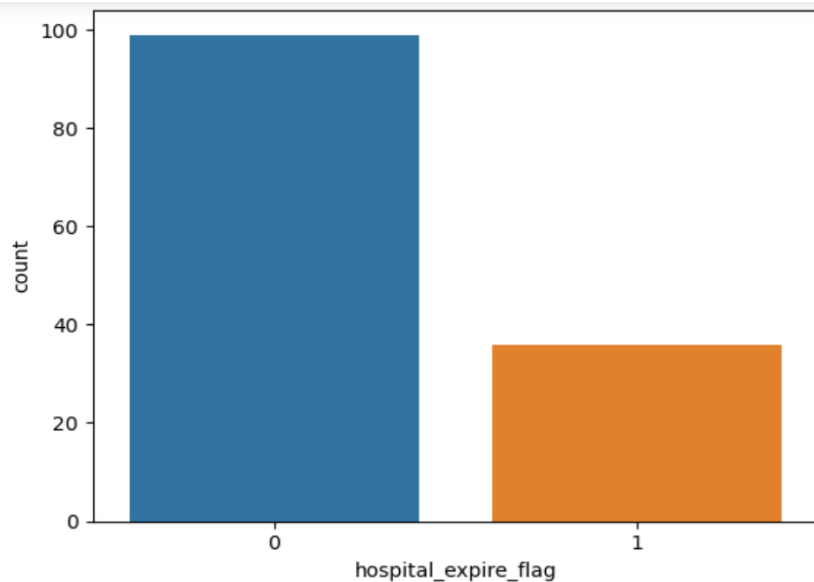


Fig.5.3.5 - Mortality Counts

Mortality count visualization - 0 represents alive count and 1 represents the death count.

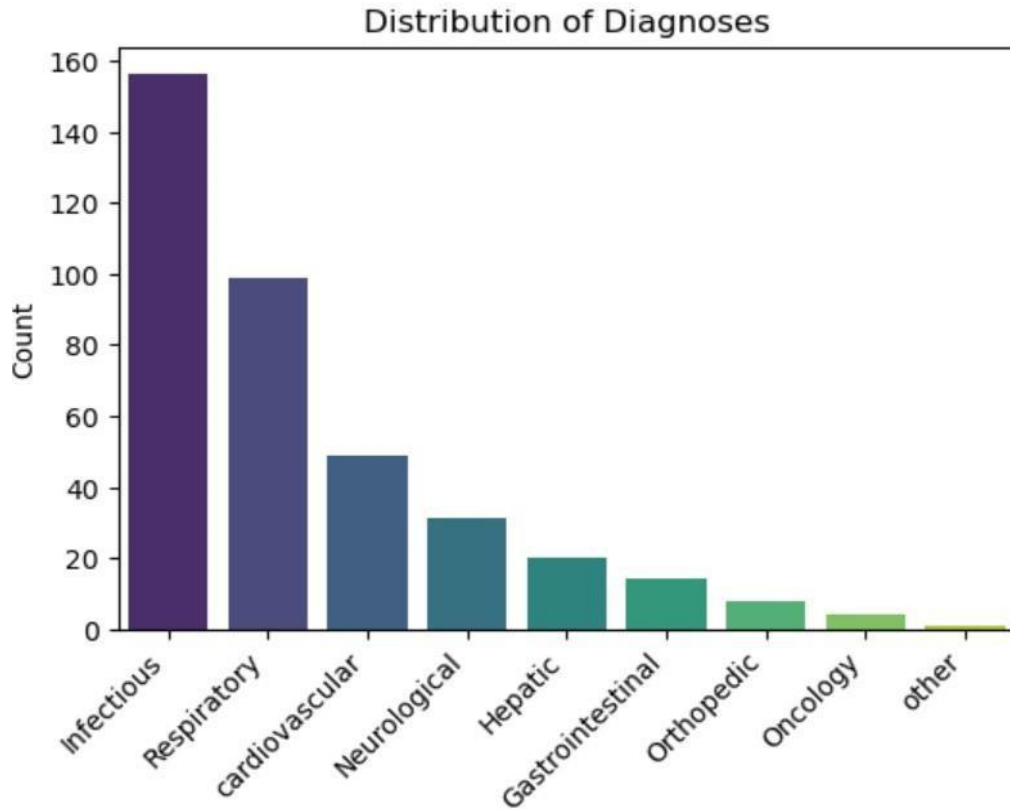


Fig.5.3.6 Diagnoses

Understanding the distribution of diagnoses among patients in Intensive Care Units (ICUs) is paramount for developing accurate and reliable mortality prediction models using machine learning algorithms. The distribution of diagnoses not only reflects the diversity of patient populations but also influences the predictive performance and generalizability of the models. In this discussion, we delve into the implications of diagnosis distribution on ICU mortality prediction and the strategies employed to address these challenges. The fig.5.3.6 shows the distribution of diagnosis from the training dataset.

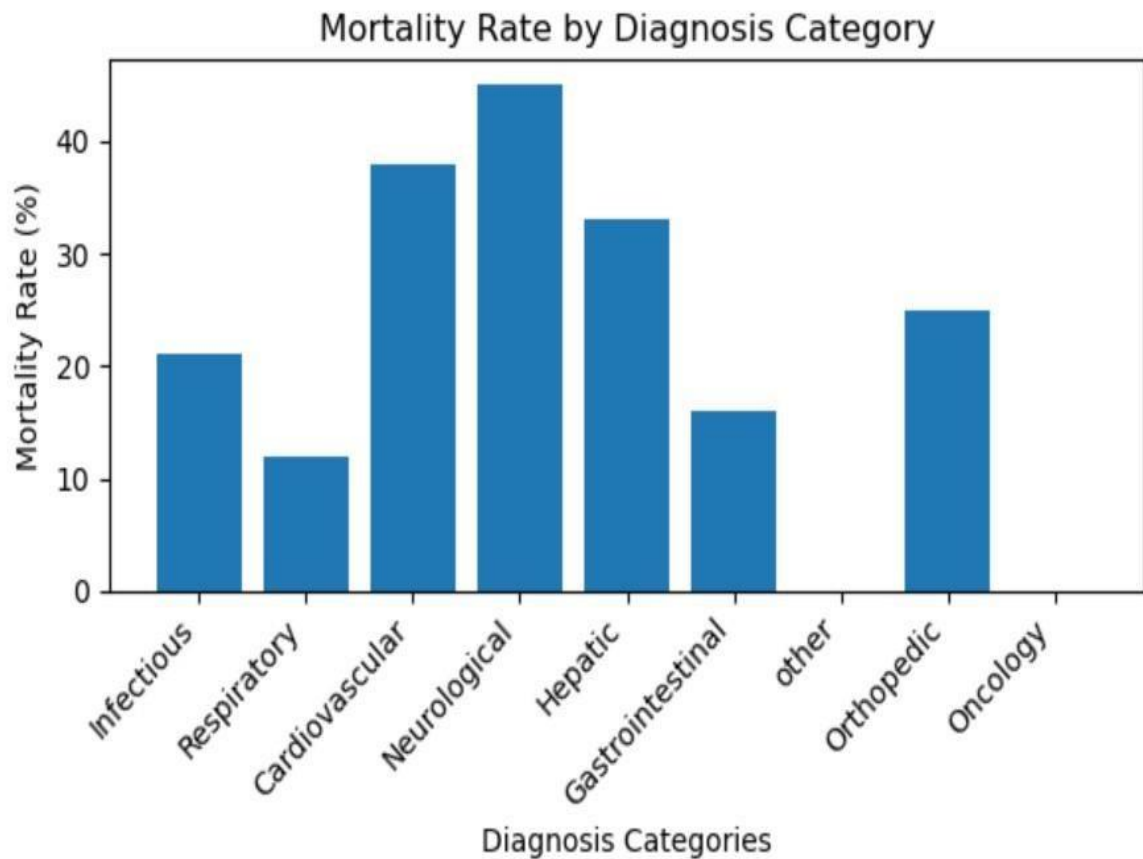


Fig.5.3.7 Mortality rate for each diagnosis

mortality rates by diagnosis are essential metrics that contribute to evidence-based healthcare decision-making, quality improvement efforts, epidemiological research, patient education, and public health planning. Analyzing mortality data in conjunction with other clinical and demographic factors provides a comprehensive understanding of disease outcomes, risk factors, and opportunities for improving healthcare delivery and patient outcomes.

5.4 FEATURE SELECTION

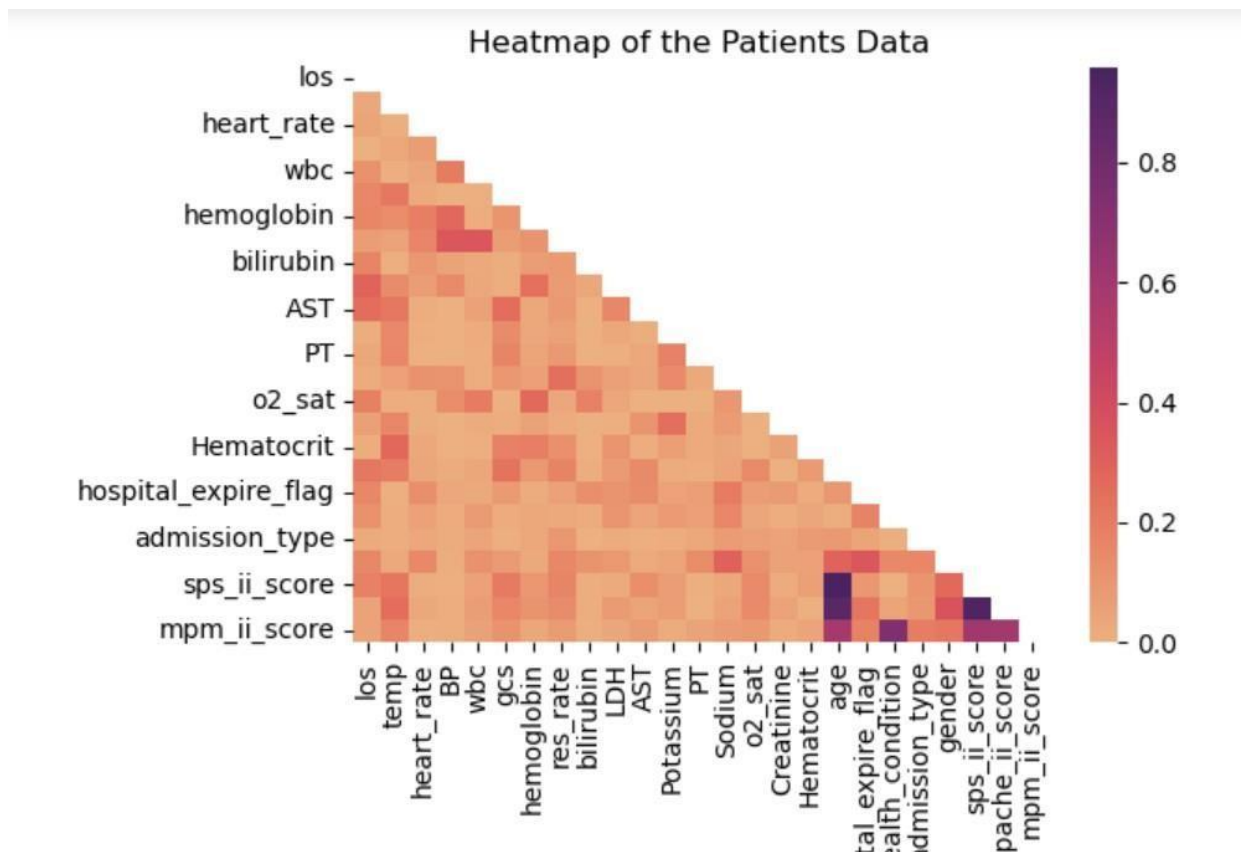


Fig.5.4.1 - Correlation Heatmap

A correlation heatmap is a graphical representation of correlation coefficients between variables in a dataset, typically displayed as a grid of coloured squares where each square represents the strength and direction of correlation between pairs of variables. The fig. 5.4.2 shows the correlation heatmap: highly correlated variables age, sps_ii_score is removed to reduce multicollinearity.

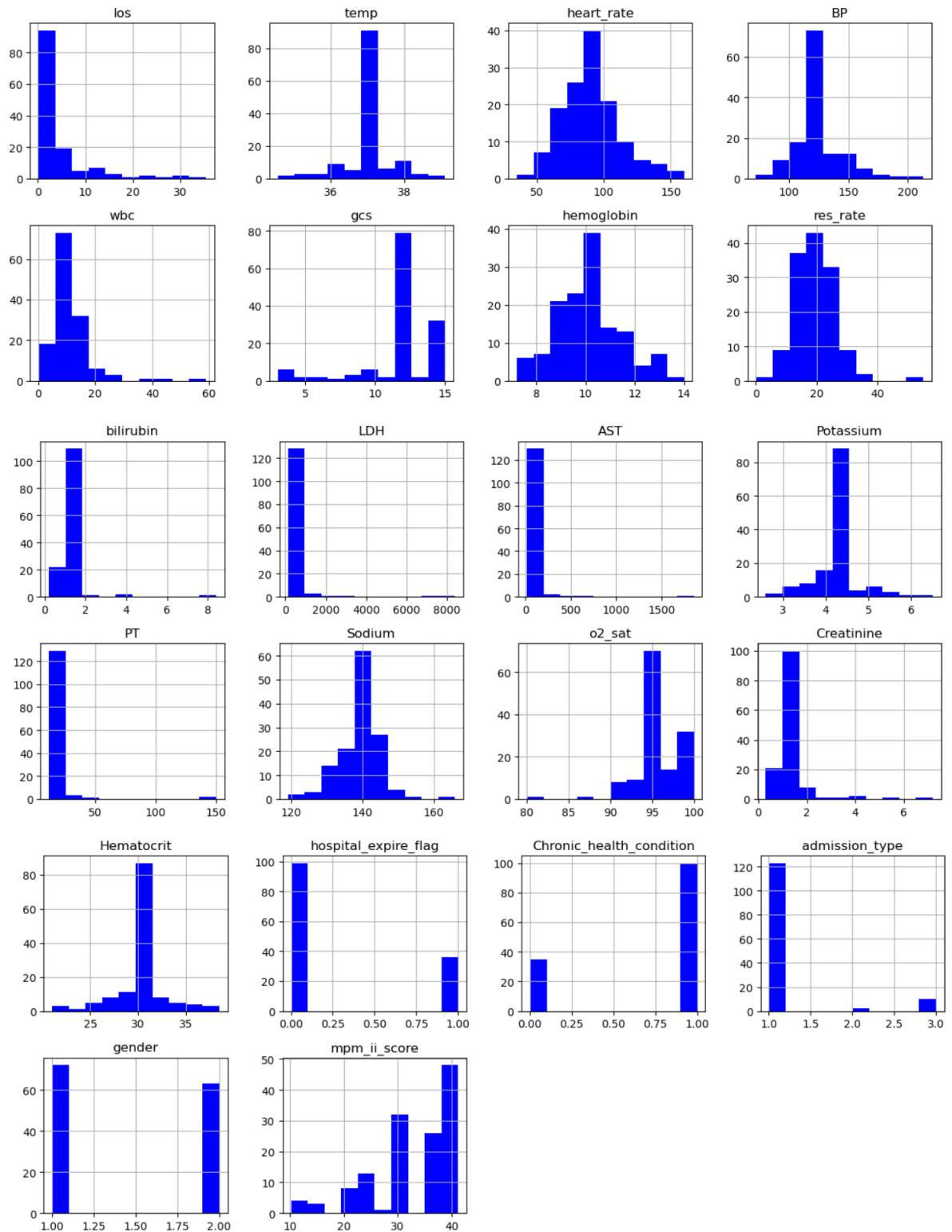


Fig .5.4.2 - Features

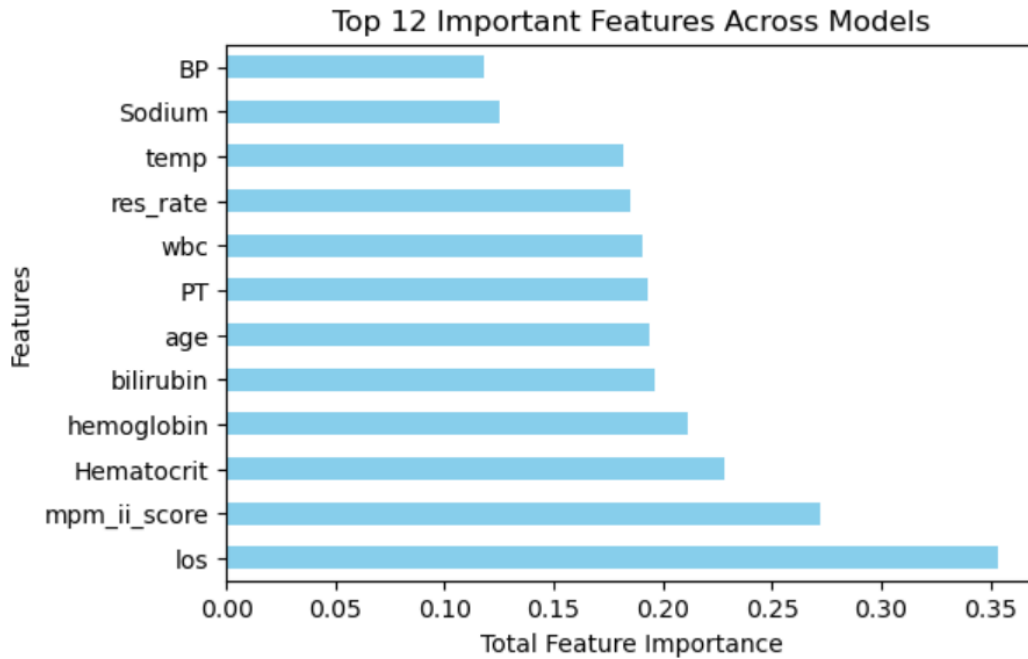


Fig .5.4.3 - Feature Importance

5.5 MODEL TRAINING

Model training discussions in ICU mortality prediction encompass a range of topics from data preparation and algorithm selection to hyperparameter tuning, evaluation metrics, interpretability, and deployment considerations. Collaborative discussions involving data scientists, clinicians, domain experts, and stakeholders are essential for designing and deploying effective predictive models that improve patient outcomes and support clinical decision-making in intensive care settings.

5.6 EVALUATION

AUROC is a valuable evaluation metric for assessing the discriminatory ability of ICU mortality prediction models. It provides a concise summary of model performance, facilitates model comparison, informs threshold selection, and complements other evaluation metrics to ensure a comprehensive assessment of model effectiveness in clinical practice.

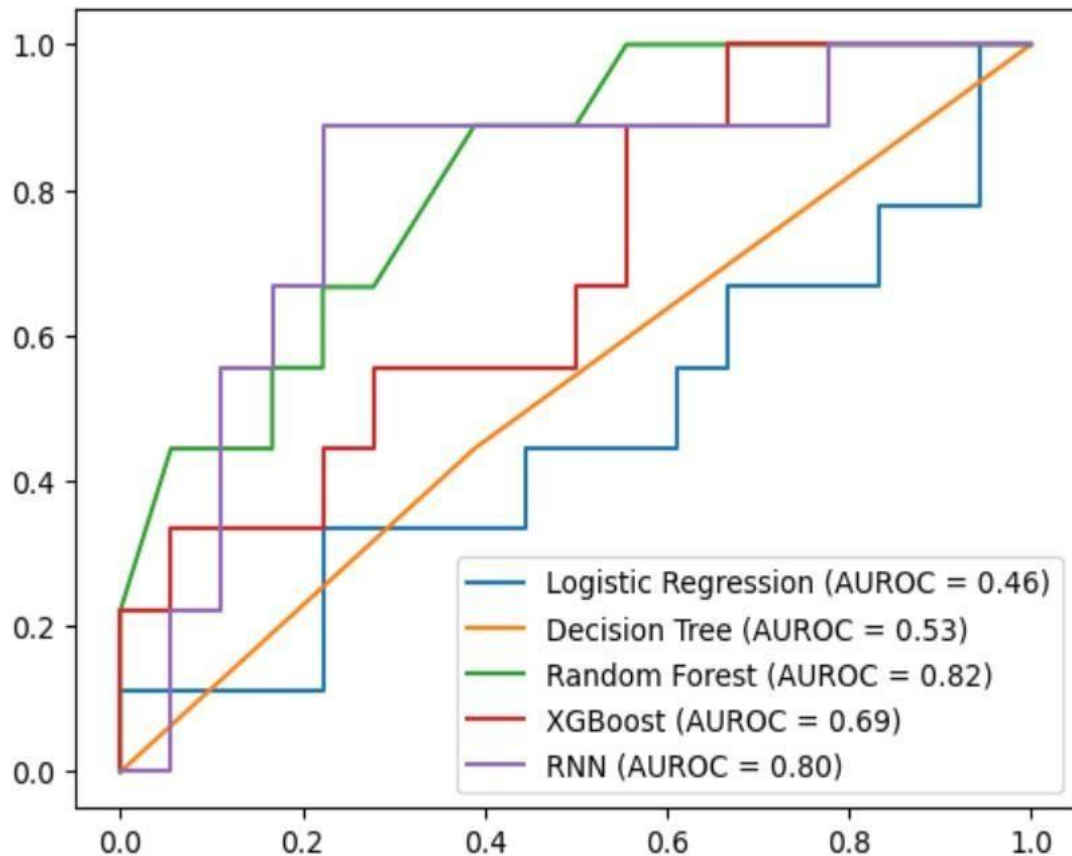


Fig.5.6.1 – AUROC Curve

Random Forest achieved an accuracy of 66% and AUC-ROC of 0.82, XGBoost showed comparable performance with an accuracy of 66%, and AUC-ROC of 0.69, Logistic Regression demonstrated good interpretability and achieved an accuracy of 67%, and AUC-ROC of 0.46, Decision Tree exhibited lower performance compared to ensemble methods with an accuracy of 59%, AUC-ROC of 0.53, Recurrent Neural Network (RNN) showed promising results with an accuracy of 70%, and AUC-ROC of 0.80, but required longer training time and computational resources.

MODELS	ACCURACY	Precision	Recall	F1 Score	AUROC SCORE
XGBoost	66%	0.50	0.33	0.40	0.69
Decision Tree	59%	0.37	0.33	0.34	0.53
Random Forest	66%	0.50	0.11	0.18	0.82
Logistic Regression	67%	1.0	0.11	0.20	0.46
Recurrent Neural Network	70%	1.0	0.11	0.20	0.80

TABLE 5.6.2: Evaluation Table

Overall, the comparative analysis highlights the trade-offs between predictive accuracy, interpretability, and computational complexity among different machine learning algorithms for ICU mortality prediction. Random Forest and RNN offer high accuracy and robustness, while Logistic Regression provides interpretability. Random Forest and RNNs have specific strengths and limitations that need to be considered based on the clinical context and data characteristics. Future research should focus on ensemble methods' interpretability, optimizing deep learning models for clinical use, and integrating machine learning tools into routine ICU care to improve patient outcomes and resource allocation.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

Mortality prediction in the ICU plays a crucial role in optimizing patient care by identifying high-risk individuals, allocating resources effectively, and tailoring treatment plans. It helps healthcare teams prioritize interventions, allocate resources efficiently, and ensure that critical resources are available where they are most needed. Additionally, mortality prediction supports personalized treatment planning and facilitates communication with patients and their families, leading to better-informed decisions and improved patient outcomes. Mortality prediction in the ICU is about leveraging data and expertise to deliver precise, compassionate, and effective care to every patient.

6.2 CONCLUSION

The main objective of this research is to provide a comprehensive approach for clinicians and managers to better manage scarce resources such as ICU beds, staff, and ventilators. Therefore, a data-driven methodology using machine learning (ML) to predict ICU mortality, and length of stay (LOS) of hospitalized patients.

To alleviate the issue of missing values, and not to delete data, the mean value is applied. Then, because of the imbalanced classes in the datasets– which degrades the prediction performance– a synthetic data generation balancing method is used to create a balanced dataset. Next, five different ML algorithms are applied, XGB, RF, Decision tree, Logistic regression, and Recurrent neural network, which are all coded in python programming language. The promising performance scores in terms of accuracy and AUC.

Our comparative analysis underscores the importance of considering trade-offs between predictive performance, interpretability, computational resources, and model complexity when selecting an algorithm for ICU mortality prediction. Each algorithm offers distinct advantages and considerations, and the choice should align with the specific needs and constraints of the healthcare environment.

Moving forward, further research is warranted to enhance the scalability, interpretability, and real-time capabilities of machine learning models in ICU mortality prediction. Incorporating domain knowledge, continuous model validation, and integration with clinical workflows will be crucial for leveraging these predictive tools effectively and ultimately improving patient outcomes in critical care settings.

REFERENCES

- Aznar-Gimeno, R, Luis M Esteban, Gorka Labata-Lezaun, Rafael Del-Hoyo-Alonso, David Abadia-Gallego (2021). A clinical decision web to predict ICU admission or death for patients hospitalized with COVID-19 using machine learning algorithms. *International Journal of Environmental Research and Public Health*, 18(16), 8677. <https://doi.org/10.3390/ijerph18168677>
- Baker S, Xiang W, Atkinson I (2020). Continuous and automatic mortality risk prediction using vital signs in the intensive care unit: a hybrid neural network approach. *Sci Rep.*10(1), 21282. <https://doi.org/10.1038/s41598-020-78184-7>.
- Bhardwaj A, Ghosh S, Gupta A, Soni BK (2021). Predictive modeling for ICU mortality prediction, A review. *Artificial Intelligence in Medicine*. 114, 102059. <https://doi.org/10.1016/j.artmed.2021.102059>
- Bihorac A, Ozrazgat-Baslanti T, Ebadi A, Motaei A, Madkour M, Pardalos PM (2019). My Surgery Risk: Development and validation of a machine-learning risk algorithm for major complications and death after surgery. *Ann Surg*. 269(4), 652-662. <https://doi.org/10.1097/SLA.0000000000002659>
- Campbell, T.W, Melissa P.W, Heinrich Roder, Samantha MaWhinney, Robert W.G, Laura K.M, Kristine M.E, Joanna Order (2021). Predicting prognosis in COVID-19 patients using machine learning and readily available clinical data. *International Journal of Medical Informatics*, 155, 104594. <https://doi.org/10.1016/j.ijmedinf.2021.104594>.
- Elhazmi, A, (2022). Machine learning decision tree algorithm role for predicting mortality in critically ill adult COVID-19 patients admitted to the ICU. *Journal of Infection and Public Health*, 15(7), 826–834. <https://doi.org/10.1016/j.jiph.2021.12.023>.
- Ghassemi M, Naumann T, Doshi-Velez F, Brimmer N, Joshi R, Rumshisky A, (2014). Unfolding physiological state: Mortality modelling in intensive care units. *KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 75-84. <https://doi.org/10.1145/2623330.2623625>

- Giulia Lorenzoni, Nicholo Sella, Annalisa Boscolo, Danila Azzolina, Patrizia Bartolotta, Laura Pasin, et al. (2021). COVID-19 ICU mortality prediction: a machine learning approach using SuperLearner algorithm. *Journal of Anesthesia, Analgesia and Critical Care*. 1(1), 3. <https://doi.org/10.1186/s44158-021-00002-x>.
- Hameed M, Alamgir Z. (2022). Improving mortality prediction in Acute Pancreatitis by machine learning and data augmentation. *Comput Biol Med*. 150,106077. <https://doi.org/10.1016/j.compbiomed.2022.106077>.
- He, Z., & Mao, J. (2020). A hybrid deep learning model for stock price prediction. *Information Fusion*, 53, 75-87. <https://doi.org/10.1016/j.inffus.2019.05.008>
- Khan, S, Michael Rieder, Safieddin Safavi-Naeini, Zsolt Jobbagy, et.al. (2021). Machine learning-based prediction of COVID-19 severity and mortality: A review. *Journal of Medical Systems*,45(1),14. <https://doi.org/10.1007/s10916-020-01677-6>
- Li, Q., Wang, K., & Yao, L. (2021). A novel intrusion detection method based on attention mechanism and long short-term memory network. *Computers & Security*, 105, 102262. <https://doi.org/10.1016/j.cose.2021.102262>
- Mahdavi, M. et al. (2021). A machine learning based exploration of COVID-19 mortality risk. *PLoS ONE*, 16(7), e0252384. <https://doi.org/10.1371/journal.pone.0252384>.
- Min Hyuk Choi, Do Kyun Kim (2022), Mortality prediction of patients in intensive care units using machine learning algorithms based on electronic health records. *Scientific Reports*. 12, 7180. <https://doi.org/10.1038/s41598-022-05910-4>
- Sara Saadatmand, Khodakaram Salimifard, Reza Mohammadi, Alex Kuiper, Maryam Marzban, Akram Farhadi (2022), Using machine learning in prediction of ICU admission, mortality, and length of stay in the early stage of admission of COVID-19 patients. *Springer Science*, 1-29. <https://doi.org/10.1007/s10479-022-04984-x>
- Shangping Zhao, Guanxiu Tang, Liu Pan, Qingyoung Wang (2023), Improving Mortality Risk Prediction with Routine Clinical Data: A Practical Machine Learning Model Based on eICU Patients. *Dovepress*. 16, 3151-3161. <https://doi.org/10.2147/CLEP.S354275>

- Strack B, DeShazo JP, Gennings C, Olmo JL, Ventura S, Cios KJ, (2014). Impact of machine learning techniques on the predictive performance of patient-level and hospital-level models of in-hospital mortality among adults admitted to the intensive care unit. PLoS ONE, 9(2), e89449. <https://doi.org/10.1371/journal.pone.0089449>
- Suparna Ghanvatkar, Vaibhav Rajan (2019), Deep Recurrent Neural Networks for Mortality Prediction in Intensive Care using Clinical Time Series at Multiple Resolutions. AIS Electronic Library (AISeL). <https://doi.org/10.17705/1Aisel.00076>
- Wang, S., Zhang, S., Li, Y., & Wu, S. (2022). A machine learning-based approach for energy consumption prediction in smart buildings. Energy and Buildings, 253, 111651. <https://doi.org/10.1016/j.enbuild.2021.111651>
- Wendong Ge, Jin-Won Huh (2018), An Interpretable ICU Mortality Prediction Model Based on Logistic Regression and Recurrent Neural Networks with LSTM units. PMCID: PMC6371274. 460–469 <https://doi.org/10.1186/s12873-018-0181-9>
- Xu, J., Xiang, L., & Liu, Q. (2021). An intelligent fault diagnosis method for rotating machinery based on convolutional neural networks and long short-term memory networks. Measurement, 169, 108431. <https://doi.org/10.1016/j.measurement.2020.108431>
- Zhang L, Huang T, Xu F (2022). Prediction of prognosis in elderly patients with sepsis based on machine learning (random survival forest). BMC Emerg Med. 22(1), 1–10. <https://doi.org/10.1186/s12873-021-00525-9>
- Zhao SP, Liu P, Tang GX (2022) External validation of a deep learning prediction model for in-hospital mortality among ICU patients. 2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA), 21–23, Shenyang, China, IEEE. <https://doi.org/10.1109/ICPECA53826.2022.9704748>

APPENDIX

SOURCE CODE

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.impute import

SimpleImputerimport numpy as np

import seaborn as sns
```

DATA COLLECTION

```
# Load the dataset

data1=

pd.read_csv("ICU

_STAYS.csv")

data2=pd.read_csv('ADMISSIONS.csv')

data3 =

pd.read_csv('PATIENTS.csv')

# Data collection

dataset = pd.merge(data1, pd.merge(data2, data3, on='subject_id', how='inner'),

on='subject_id', how='inner')

dataset.head()

dataset =

dataset.drop_duplicates(subset=['icustay_id'])

len(dataset)
```

DATA PREPROCESSING

```
dataset.isnull().sum()

print("Basic Information about the Dataset:")

print(dataset.info())

# Display summary statistics of numerical columns

print("\nSummary Statistics:")

print(dataset.describe())

# gender

gender_mapping = {'F': 2, 'M': 1}

admission_type_mapping = {'EMERGENCY': 1, 'URGENT': 2, 'ELECTIVE': 3}

# Apply the mapping function

dataset['gender'] = dataset['gender'].map(gender_mapping) dataset['admission_type'] =
dataset['admission_type'].map(admission_type_mapping)dataset.head()
```

DATA VISUALIZATION

```
sns.countplot(data=final_data, x="hospital_expire_flag")

gender_counts = dataset['gender'].value_counts()

gender_counts.plot(kind='bar', color=['skyblue', 'pink'])

corr_matrix = np.abs(final_data.corr())

mask = np.triu(np.ones_like(corr_matrix) > 0)

plt.figure(figsize = (6, 4))

sns.heatmap(corr_matrix, mask=mask, cmap='flare')

plt.title('Heatmap of the Patients Data')

plt.show()

final_data = final_data.drop(['age', 'sps_ii_score', 'apache_ii_score'], axis=1)
```

```

final_data.hist(figsize=(15,50), layout = (15,4), color = 'blue')

plt.show()

# Define categories based on keywords

categories = {

    'cardiovascular': ['REGURGITATION',
'STENOSIS','ARTERY','ARREST','STROKE','HYPOTENSION','ANGINA','PULMONA
RY','BRADYCARDIA','MI', 'PERICARDIAL','MYOCARDIA'],

    'Gastrointestinal': ['CHOLANGITIS',
'CHOLECYSTITIS','ESOPHAGEAL','ABDOMINAL','VOLVULUS'],

    'Neurological': ['ALCOHOLIC', 'MENTAL', 'AROMEGLEY', 'LYMPHOMA',
'NUMBNESS','HEADACHE', 'METASTASES',

                    'ACCIDENT', 'STENOSIS' 'HEMATOMA', 'SEIZURE',
'SYNCOPE','HEMORRHAGE','FAILURE', 'SYNOVIAL'],

    'Infectious': ['ABSCESS','CELLULITIS','FEVER','SEPSIS', 'UTI', 'UROSEPSIS',
'HYPONATREMIA','INFECTION','PYELONEPHRITIS','TRACHEAL'],

    'Respiratory': ['ACUTE', 'EMBOLISM', 'ASTHMA', 'DISEASE','TACHYPNEA',
'BREATH', 'RESPIRATORY', 'EFFUSION', 'PNEUMONIA', 'TELEMETRY'],

    'Hepatic': ['HEPATITIS', 'HEPATIC','LIVER','FISTULA', 'SDA', 'PAIN', 'BLEED'],

    'Oncology': ['AROMEGLEY','METASTASIS', 'METASTIC', 'ANEMIA',
'ADENOPATHY', 'CANCER'],

    'Orthopedic': ['FRACTURE', 'FALL']

    # Add more categories and keywords as needed

}

# Function to categorize diagnoses based on keywords

def categorize_diagnosis(diagnosis):

    for category, keywords in categories.items():

        for keyword in keywords:

```

```

        if keyword in
            diagnosis:return
            category

    return 'other' # Default category if no match
found# Apply categorization to diagnoses
dataset['diagnosis_category'] =
dataset['diagnosis'].apply(categorize_diagnosis)# Display categorized
diagnoses
print(dataset[['diagnosis',
'diagnosis_category']])# Explore the
distribution of diagnoses
diagnosis_counts =
dataset['diagnosis_category'].value_counts()# Plot diagnosis
distribution
plt.figure(figsize=(6, 4))
sns.barplot(x=diagnosis_counts.index, y=diagnosis_counts.values, palette='viridis')
plt.title('Distribution of Diagnoses')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.xticks(rotation=45,
ha='right')plt.show()
print(diagnosis_counts)

# Calculate mortality count for each category
mortality_counts =
dataset.groupby('diagnosis_category')['hospital_expire_flag'].sum()

```

```

print(mortality_counts)

cat_data = {
    'Diagnosis_Category': ["Infectious", "Respiratory", "Cardiovascular",
                           "Neurological", "Hepatic", "Gastrointestinal", "other", "Orthopedic", "Oncology"],
    'Total_Patients': [156, 99, 49, 31, 20, 14, 1, 8, 4],
    'Total_Deaths': [7, 4, 12, 16, 8, 4, 0, 2, 0],
}

# Create DataFrame
data = pd.DataFrame(cat_data)

# Calculate mortality rate for each category
data['Mortality_Rate'] = (data['Total_Deaths'] / data['Total_Patients']) * 100

# Display DataFrame with mortality rates
print(data)

diagnosis_categories = ["Infectious", "Respiratory", "Cardiovascular", "Neurological",
                        "Hepatic", "Gastrointestinal", "other", "Orthopedic", "Oncology"]

mortality_rates = [4, 4, 24, 51, 40, 28, 0, 25, 0] # Mortality rates for each category

# Plotting
plt.figure(figsize=(6, 4))

plt.bar(diagnosis_categories, mortality_rates)

plt.xlabel('Diagnosis Categories')

plt.ylabel('Mortality Rate (%)')

plt.title('Mortality Rate by Diagnosis Category')

plt.xticks(rotation=45, ha='right')

plt.tight_layout()

plt.show()

diagnoses = {

```

```

    'Chronics':['cardiovascular','Neurological','Hepatic','Oncology','Respiratory'],

    'Others': ['Gastrointestinal', 'Infectious', 'Orthopedic', 'other']

}

def cat_diagnosis(diagnosis):

    for category, keywords in diagnoses.items():

        for keyword in keywords:

            if keyword in diagnosis:

                return category

    return 'other'

# Apply categorization to diagnoses

dataset['Chronic_type'] = dataset['diagnosis_category'].apply(cat_diagnosis)

print(dataset[['diagnosis_category', 'Chronic_type']])

dataset = dataset.drop(dataset[dataset['has_chartevents_data'] == 0].index)

print(len(dataset))

```

FEATURE SELECTION

```

feat_data=dataset

[['los',"temp","heart_rate","BP","wbc","gcs","hemoglobin","res_rate","bilirubin","LDH",
"AST","Potassium","PT","Sodium","o2_sat","Creatinine","Hematocrit","age","hospital_
expire_flag", "Chronic_health_condition","admission_type","gender"]]

feat_data.tail()

final_data = feat_data.fillna(feat_data.mean())

final_data.tail()

def calculate_sps_ii(age, BP, heart_rate, temperature, respiratory_rate, sodium,
potassium, creatinine, hematocrit, wbc, gcs):

    sps_ii_score = 0

```

```

# Calculate score based on age
if age >= 70:

    sps_ii_score += 11

elif 60 <= age < 70:

    sps_ii_score += 9

elif 50 <= age < 60:

    sps_ii_score += 8

elif 40 <= age < 50:

    sps_ii_score += 7

elif 30 <= age < 40:

    sps_ii_score += 5

elif 20 <= age < 30:

    sps_ii_score += 3

elif age < 20:

    sps_ii_score += 0

# Add other components and their respective weights

return sps_ii_score

# Function to calculate SPS II score for a single row (patient)

def calculate_sps_ii_row(row):

    sps_ii_score = calculate_sps_ii(row['age'], row['BP'], row['heart_rate'], row['temp'],
row['res_rate'], row['Sodium'], row['Potassium'], row['Creatinine'], row['Hematocrit'],
row['wbc'], row['gcs'])

    return sps_ii_score

# Calculate SPS II score for each patient

final_data['sps_ii_score'] = final_data.apply(calculate_sps_ii_row, axis=1)

```

```

final_data.head()

def calculate_apache_ii(age, temperature, mean_arterial_pressure, heart_rate,
respiratory_rate, oxygenation, sodium, potassium, creatinine, hematocrit,
white_blood_cell_count, Glasgow_coma_score):

    apache_ii_score = 0

    # Calculate score based on age

    if age >= 70:

        apache_ii_score += 5

    elif 60 <= age < 70:

        apache_ii_score += 3

    elif 50 <= age < 60:

        apache_ii_score += 2

    elif 40 <= age < 50:

        apache_ii_score += 1

    # Calculate score based on temperature

    if temperature < 34:

        apache_ii_score += 4

    elif 34 <= temperature < 35:

        apache_ii_score += 3

    elif 35 <= temperature < 36:

        apache_ii_score += 1

    elif 38 <= temperature < 39:

        apache_ii_score += 1

    elif temperature >= 39:

```

```

    apache_ii_score += 2

# Add other components and their respective weights

return apache_ii_score

def calculate_apache_ii_row(row):

    apache_ii_score = calculate_apache_ii(row['age'], row['temp'], row['BP'],
row['heart_rate'], row['res_rate'], row['o2_sat'], row['Sodium'], row['Potassium'],
row['Creatinine'], row['Hematocrit'], row['wbc'], row['gcs'])

    return apache_ii_score

final_data['apache_ii_score'] = final_data.apply(calculate_apache_ii_row, axis=1)

final_data.tail()

def calculate_mpm_ii(age, acute_physiology_score, chronic_health_condition,
admission_type):

    mpm_ii_score = 0

    # Calculate score based on age

    if age >= 70:

        mpm_ii_score += 20

    elif 60 <= age < 70:

        mpm_ii_score += 15

    elif 50 <= age < 60:

        mpm_ii_score += 10

    elif 40 <= age < 50:

        mpm_ii_score += 5

    mpm_ii_score += acute_physiology_score * 0.2

    mpm_ii_score += chronic_health_condition * 10

    if admission_type == 1:

```

```

        mpm_ii_score += 10

    elif admission_type == 2:

        mpm_ii_score += 8

    elif admission_type == 0:

        mpm_ii_score += 5

    # Add additional adjustments based on admission diagnosis or other factors

    return mpm_ii_score

# Function to calculate SPS II score for a single row (patient)

def calculate_mpm_ii_row(row):

    mpm_ii_score = calculate_mpm_ii(row['age'], row['apache_ii_score'],
    row['Chronic_health_condition'],row['admission_type'])

    return mpm_ii_score

# Calculate SPS II score for each patient

final_data['mpm_ii_score'] = final_data.apply(calculate_mpm_ii_row, axis=1)

final_data.tail()

final_data.to_csv('Features.csv', index=False)

final_data.tail()

```

MODEL TRAINING

```

from sklearn import metrics

from sklearn import datasets

from sklearn.preprocessing import StandardScaler

from xgboost import XGBClassifier

from sklearn.model_selection import train_test_split

```

```

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from tensorflow import keras

from tensorflow.keras import layers

from keras.models import Sequential

from keras.layers import Dense, GRU, TimeDistributed

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_auc_score

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

X = data.drop('hospital_expire_flag', axis=1) # Features
y = data['hospital_expire_flag']

#split dataset into training and testing set

X_train,X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,random_state=42)

models = {

    'Random Forest': RandomForestClassifier(),

    'Decision Tree': DecisionTreeClassifier(),

    'Logistic Regression': LogisticRegression(),

    'XGBoost': XGBClassifier()

}

# Add RNN model

rnn_model = Sequential([

    GRU(15,      activation='tanh',      recurrent_activation      =      "sigmoid",
input_shape=(X_train.shape[1], 1)),

```

```

        Dense(1)

    ])

    rnn_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

    # Reshape data for RNN input

    X_train_rnn = np.expand_dims(X_train, axis=-1)

    X_test_rnn = np.expand_dims(X_test, axis=-1)

    # Train RNN model

    rnn_model.fit(X_train_rnn, y_train, epochs=100, batch_size=32, verbose=2)

```

EVALUATION

```

# Evaluate RNN model

y_proba_rnn = rnn_model.predict(X_test_rnn)

y_pred_rnn = (y_proba_rnn > 0.5).astype(int).flatten()

accuracy_rnn = accuracy_score(y_test, y_pred_rnn)

precision_rnn = precision_score(y_test, y_pred_rnn)

recall_rnn = recall_score(y_test, y_pred_rnn)

f1_rnn = f1_score(y_test, y_pred_rnn)

auroc_rnn = roc_auc_score(y_test, y_proba_rnn)

print(precision_rnn, recall_rnn)

# Standardize features

models['RNN'] = rnn_model

# Create a list to store evaluation metrics

metrics_list = []

# Evaluate each model and store results in the list

```

```

for name, model in models.items():

    if name == 'RNN':

        accuracy = accuracy_rnn

        precision = precision_rnn

        recall = recall_rnn

        f1 = f1_rnn

        auroc = auroc_rnn

    else:

        model.fit(X_train, y_train)

        y_pred = model.predict(X_test)

        y_proba = model.predict_proba(X_test)[:, 1] # Probability of positive class

        accuracy = accuracy_score(y_test, y_pred)

        precision = precision_score(y_test, y_pred)

        recall = recall_score(y_test, y_pred)

        f1 = f1_score(y_test, y_pred)

        auroc = roc_auc_score(y_test, y_proba)

    metrics_list.append({

        'Model': name,

        'Accuracy': accuracy,

        'Precision': precision,

        'Recall': recall,

        'F1 Score': f1,

        'AUROC': auroc

    })

```

```

# Convert the list of dictionaries to a DataFrame
metrics_df = pd.DataFrame(metrics_list)

print(metrics_df)

plt.figure(0).clf()

# Evaluate each model and store results in the list
for name, model in models.items():

    if name == 'RNN':

        y_proba_rnn = rnn_model.predict(X_test_rnn)

        y_pred_rnn = (y_proba_rnn > 0.5).astype(int).flatten()

        rnn_fpr, rnn_tpr, _ = metrics.roc_curve(y_test, y_proba_rnn)

        rnn_auc = round(metrics.roc_auc_score(y_test, y_proba_rnn), 4)

        plt.plot(rnn_fpr, rnn_tpr, label=f'{name} (AUROC = {rnn_auc:.2f})')

    else:

        # Probability of positive class

        y_pred = model.predict_proba(X_test)[:, 1]

        fpr, tpr, _ = metrics.roc_curve(y_test, y_pred)

        auc = round(metrics.roc_auc_score(y_test, y_pred), 4)

        plt.plot(fpr, tpr, label=f'{name} (AUROC = {auc:.2f})')

plt.legend()

```