Predicting Mortality of Intensive Care Patients with Machine Learning Using Electronic Health Record and Non-Invasive Signals

Ali Neissi Shooshtari

Master's Thesis
Master of Engineering - Big Data Analytics
July 23, 2021

MASTER'S THESIS	
Arcada University of Applied Sciences	
Degree Programme:	Master of Engineering - Big Data Analytics
Identification number:	8271
Author:	Ali Neissi Shooshtari
Title:	Predicting Mortality of Intensive Care Patients with Machine Learning Using Electronic Health Record and Non-Invasive Signals
Supervisor (Arcada):	Leonardo Espinosa Leal
Commissioned by:	GE Healthcare

Abstract:

This study proposes a novel approach for applying the Electronic Health Record (EHR) data and biomedical signals to predict patient mortality in the Intensive Care Unit (ICU) using machine learning and deep learning models. The study results showed that using a combination of EHR data and waveforms improved the performance of the models compared to using only one of the inputs. A dataset containing 2320 ICU patients was used in this study. A 5-hour data extraction window for EHR data and a 1-hour data extraction window for the waveform were considered. The prediction window was set to 12 hours. Five different models were used in this study. Six vitals along with three non-invasive signals were used and in total, some 67 different features were extracted and used. Area Under Receiver Operating Characteristic Curve (AUROC) and Area Under Precision Recall Curve (AUPRC) were taken as the metrics. The best performance achieved in this study using both EHR data and waveforms was an AUROC of 0.877 and an AUPRC of 0.289. The results also showed the model fed by a combination of the EHR data and waveforms outperformed the same models when they were fed only with EHR data by 3% in terms of AUROC and by 10.3% in terms of AUPRC and, the models that were fed only by waveforms by 4.4% in terms of AUROC and 7.03% in terms of AUPRC.

Keywords:	Electronic Health Record, Intensive Care Unit, Machine Learning, Deep Learning, Mortality
Number of pages:	72
Language:	English
Date of acceptance:	25.05.2021

CONTENTS

FIGURES

TABLES

ABBREVIATIONS

APACHE	Acute Physiology and Chronic Health Evaluation
AI	Artificial Intelligence
AUC	Area Under Curve
AUPRC	Area Under Precision Recall Curve
AUROC	Area Under Receiver Operating Characteristic Curve
AWS	Amazon Web Services
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
EC2	Amazon Elastic Compute Cloud
ECG	Electrocardiogram
EHR	Electronic Health Record
EWS	Early Warning Score
XGB	Extreme Gradient Boosting
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GB	Gradient Boosting
IBP	Invasive Blood Pressure
ICU	Intensive Care Unit
IP	Impedance Pneumogram
LOD	Logistic Organ Dysfunction
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
MODS	Multiple Organs Dysfunction Score
NIBP	Non-Invasive Blood Pressure
PPG	Photoplethysmogram
PRC	Precision Recall Curve
RF	Random Forest

RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic Curve
S3	Amazon Simple Storage Service
SOFA	Sequential Organ Failure Assessment
SpO2	Peripheral Oxygen Saturation
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

FOREWORD

I would like to appreciate Hanna Viertiö-Oja and GE Healthcare Finland for accommodating me to work within the Data Science team of Acute Care towards writing my master's thesis.

I want to extend my deepest gratitude to my advisors Joanna Haahti and Jyri Kivinen to share their extensive knowledge with me, and support me by all means. Without them, this thesis work was not possible.

I want to thank Magnus Westerlund for his support in the initial phase of the thesis that helped me to frame the problem better. I also want to thank my supervisor Leonardo Espinosa Leal for his invaluable comments, tips, and feedback, especially during the writing part.

It was truly an honor for me to be able to learn from such wonderful experts in the field that I am passionate about.

Finally, I would like to thank the dearest family of mine and all of my dear friends all around the world for supporting me throughout the whole process of writing this thesis with a special thanks to Ali Talasaz who is like a brother to me, and the wonderful Liza Rossi for sharing with me, her vast knowledge and experience of working as a doctor in the Intensive Care Unit.

1 INTRODUCTION

The ability to accurately predict the situation of the hospitalized patient in the future brings invaluable information both for enhancing patient care and more efficient use of hospital resources. One of these situations of interest is patient deterioration and its early detection. As a response to this need different scoring systems have been developed to analyze the severity of a patient's condition inside and outside the Intensive Care Unit (ICU) (Morgan et al. 1997, Rapsang & Shyam 2014).

The scoring systems, despite providing good information, have some limitations. They are calculated based on a few variables and hence can not consider each patient's characteristics separately. That is why critical evaluation of these scores is needed before any decisions towards treatment (Keegan et al. 2011). Furthermore, these scores are calculated based on the patient's situation at that specific time and therefore can not take into account the temporary changes of the variables (Bouch & Thompson 2008).

The emergence of Electronic Health Record (EHR) systems in which the medical data could be stored and accessed, enabled more advanced Machine Learning (ML) and Deep Learning (DL) methods to be used in prediction tasks in healthcare such as prediction of patient deterioration, length of stay in the hospital, readmission and other clinical events (Bright et al. 2012, Bronzino & Peterson 2014).

EHR systems contain information on patients' vital signs such as heart rate, respiration rate, blood pressure, and oxygen saturation that are calculated from Electrocardiogram (ECG), Impedance Pneumogram (IP), Invasive Blood Pressure (IBP), and Photoplethysmogram (PPG) biomedical signals. These signals are monitored by the bedside monitoring solution devices but they are not stored in the EHR themselves. laboratory test results, list of allergies, medication, and doctors' notes on the patient are other useful information that is stored and can be found in the EHR as well (Aspden 2004).

The main input to scoring systems and computer-based prediction systems has been the EHR data (Bright et al. 2012). The biomedical signals have not been utilized in the prediction tasks. The first study for uncovering the potentials of biomedical signals in predicting patients deterioration was done by Haahti (2019) where she used long sequences of waveform signals as the input to the predictive models to explore the potentials of the signals in predictive tasks in clinical decision making.

These signals have a raw format and they are large in size, therefore they are challenging for humans to analyze. That is why providing a digestible output from these signals after being analyzed by deep learning models would give value to the data that a human can't easily analyze.

This study is the continuation of the work done by Haahti (2019) in which a novel approach is implemented to combine the usage of EHR data and biomedical waveform as inputs to predictive models.

Haahti (2019) had clearly defined deterioration as mortality to start with and formulated the prediction task as a binary classification using three non-invasive signals (ECG, IP, and PPG) as inputs to neural network models. These three signals are concurrent temporal sequences that represent the patient's state.

The prediction models utilize one hour of ECG, IP, and PPG signal as the input to predicting models, resulting in very large input sizes that create challenges for deep learning models. However, there are characteristics in these signals such as arrhythmia that are not present in the EHR data (Haahti 2019).

In this study, six different vital signs (Heart Rate, Respiration, Systolic Blood Pressure, Diastolic Blood Pressure, Pulse Oximetry, and Temperature) from EHR data are used. For each vital sign, eleven different statistics are extracted from the EHR data. Then a few suitable models for binary classification of structured data are fit to the data and the model with the best performance is selected.

The deep learning model developed by Haahti (2019) is trained with the cohort of this study and then prediction probabilities gained from that model is fed as an extra feature to the best model of the former step.

The thesis includes the following chapters. This chapter continues by explaining the vital signs. In Chapter 2 the related work done formerly is discussed. In chapter 3 the dataset used in this study is explained. In chapter 4, the methodologies used in the experiments of this thesis are described. Chapter 5 details the experiments conducted in this thesis and Chapter 6 includes the results. In Chapter 7 the methods and the results of this study are discussed in comparison to the former relevant research. Finally, in chapter 8 thesis summary and the conclusions are given.

1.1 Background

In this section, first, patient monitoring in the ICU is explained. Then some ideas about the prediction of patient deterioration in the ICU are discussed. And finally, the vital signs used in this study and their corresponding biomedical signals are explained.

1.1.1 Patient Monitoring in the ICU

The need for improving patient records in the hospitals was first addressed in 1991 in the General Accounting Office (GAO) report (Dick et al. 1997). These systems were referred to as computer-based patient records. In which three major benefits of implementing an automated medical reporting system for health care were identified.

First, improvement in data access, faster data retrieval, higher quality data, and more versatility in data display would enhance healthcare delivery. Second, electronically captured information could facilitate research programs and help for better outcomes. Third, automated patient records could reduce cost and enhance the productivity of the staff which could lead to higher hospital efficiency (Dick et al. 1997).

Later in 2003, the U.S. Institute of medicine defined an EHR system as "an EHR system includes longitudinal collection of electronic health information for and about persons, where health information is defined as information pertaining to the health of an individual or health care provided to an individual" (Institute of Medicine (US) 2003).

The patients who are admitted to ICU are usually in very critical situations (Nates et al. 2016). In the ICU usually, a set of probes are connected to the patient, through those the biomedical signals are captured. Three of these signals are ECG, IP, and PPG. From these signals, the vital signs such as Heart rate, respiratory rate, and oxygen saturation are calculated and recorded in the EHR with certain frequencies. Furthermore, EHR includes other information such as comments written by doctors and nurses about the patient, patient's laboratory test results, medication lists, and possible allergies (Medicine et al. 2003). The waveforms of biosignals are not saved in the EHR.

1.1.2 Prediction of patient mortality

The EHR data is structured data that is derived through rule-sets from the waveforms. In this process, a lot of information from the waveforms is neglected. Therefore, the aim is to study if waveforms as unstructured data can improve the performance of the models when used as a combination with EHR data. As the initial step, we consider death as deterioration because labeling the death is simple as it is clear what has happened before and after the event. Later a similar approach can be used for predicting other deterioration, organ failures, etc. Here the idea is that if using a combination of the waveform and EHR outperforms the use of either EHR data or waveforms.

It might be perceived that predicting the deterioration might be a lot more useful in the ward than in the ICU. Because in the ward the number of caregivers is less and this information in the ward can help in bringing more attention to the patients with a higher chance of deterioration as well as helping the care providers in decision making whether to transfer the patient to the intensive care or for example decide if surgery is needed.

However due to a lot more frequent measurements in the ICU, the amount of data generated there is much larger compared to the ward and high care, so that is why the ICU has been the interest of machine learning communities to conduct their research and make their algorithms for the ICU. Besides there has been an interest in the ICU to evaluate the severity of a patient's illness, sometimes referred to as mortality based on their test results within 24 hours of their admission even before machine learning has kicked in.

Several scoring systems such as Simplified Acute Physiologic Score (SAPS II), Multiple Organ Dysfunction Syndrome (MODS), Acute Physiology and Chronic Health Evaluation (APACHE II), Sequential Organ Failure Assessment (SOFA), Logistic Organ Dysfunction (LOD), and several others have been developed and have been used with certain accuracies (Haddadi et al. 2014, Baue et al. 1998). These scoring systems aimed to classify patients into different groups based on illness severity or risk of mortality. So, as it is perceived in this track of research the value proposition is bringing attention and optimizing the use of hospital resources. This can help in better decision-making, giving the care at the right time, and reducing hospital costs (Bates et al. 2014).

The machine learning algorithms developed for the ICU are to replace these scoring systems with the hope of bringing more ease of use and better accuracy. Current algorithms that do the scoring tend to predict the overall situation of a patient. The protocol of giving care is not altered but by classifying the patients, the aim is to improve productivity and this is helping the patients indirectly.

However, what if we would like to predict what happens to the patient in a few hours and aim to help the patient directly. Meaning an action in the form of treatment can be done by the health providers based on this prediction. For example, the algorithm can predict a patient's deterioration and specify the reasons for that. Or specifically, predict an organ failure providing the reasons. In this case, the value proposition can be defined as directly helping the patient.

Doing this requires a lot more effort to understand and consider the caregivers' thoughts in the phases of making the technology from brainstorming to the product release.

In the following, the five vital signs and their corresponding biological signals are explained to give an overview.

1.2 Vital Signs

The main vital signs that are measured by health professionals are Pulse rate, respiratory rate, blood pressure, Pulse Oximetry, and body temperature. In this chapter, these vital signs are briefly covered. In the ICU these vital signs are measured by bedside monitoring devices. In the figure 1 a bedside monitoring system in the ICU is shown. More specifically the sensor probes attached to the patient's body sense the signals and send them to the acquisition modules connected to the patient monitors. After the signals are processed the vital signs signals and values are shown on the monitors' screen.

Figure 1. A monitoring solution system in the ICU (Gehealthcare.com 2021)

1.2.1 Pulse or Heart Rate

Pulse or heart rate is the number of times one's resting heart beats per minute. The usual range for heart rate is between 60 and 100 but the heart rate can change minute to minute. While exercising the heart rate can reach 130-150 beats per minute (Åstrand & Ryhming 1954).

In the intensive care, the heart rate is calculated from monitoring the electrocardiogram (ECG or EKG). The standard form of ECG consists of 12 leads. In the hardwire ECG, electrode pads are attached to the patient's chest. Then lead cables are connected to these pads from one side and to the bedside monitors from the other side. Out of the 12 leads in standard ECG, six of them are placed on the legs and arms. These leads are called "limb leads". The other 6 leads are placed on the precordium and they are called "precordial leads" (Cadogan 2021). Figure 2 shows the placement of ECG leads on the body.

There are several methods for calculating the heart rate from the ECG. One of them is the square count method. There is a grid on the paper on which the ECG is printed. This grid is in form of small and large squares. Every 5x5 small squares make a large square. In the square count method, the sequence of 300-150-100-75-60-50-43-38-33 is followed.

When the signal is recorded with the speed of 25 mm/s, if the distance between two peaks in ECG is one big square then the heart rate is 300 bpm. If the distance between two peaks

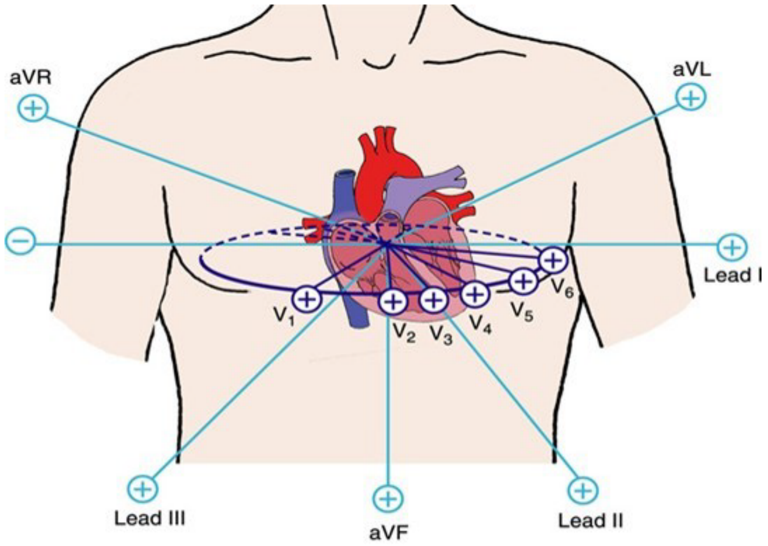


Figure 2. The locations of leads in a 12-lead ECG each giving a different view of the hearts electrical activity. The leads are divided into three groups: six precordial leads (V1, V2, V3, V4, V5, V6), three limb leads (I, II, III), and three augmented limb leads (aVR, aVL, aVF) (Randazzo 2016)

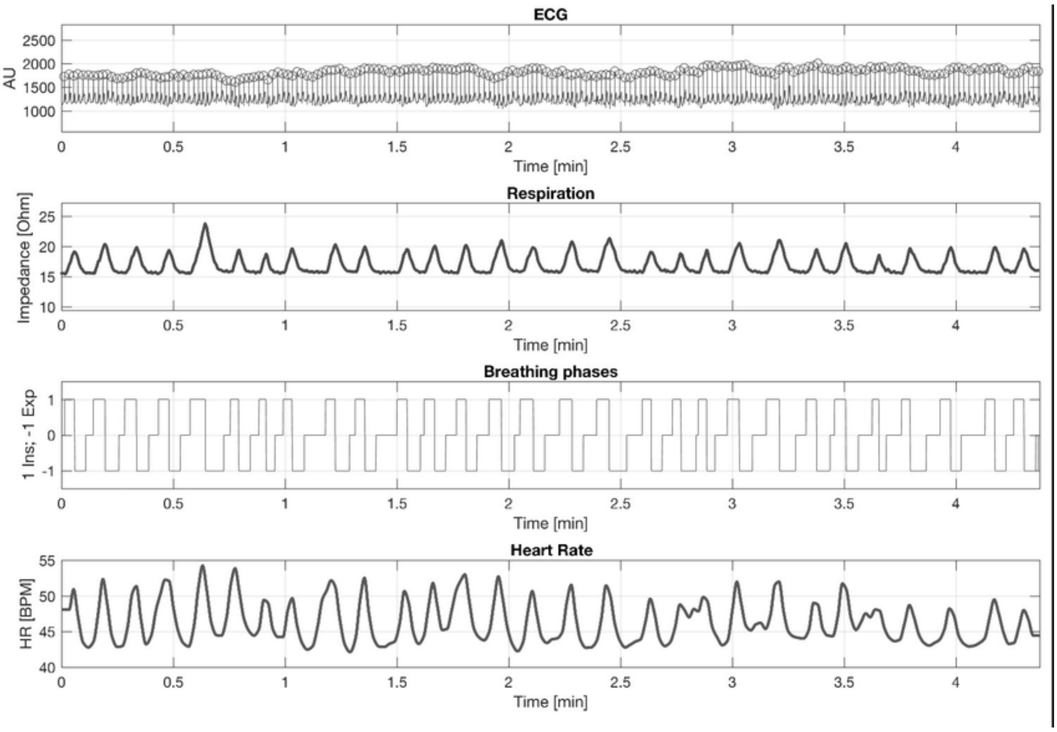


Figure 3. Sample ECG and IP signals and correspondingly calculated respiration phases and tachogram (Młyńczak et al. 2017)

is two big squares, then the heart rate is 150 bpm. If the distance between two peaks is five large squares then the heart rate is 60 bpm (Proven 2019).

The second method would be counting the small squares. When signal is recorded with the speed of 25 mm/s, then the heart rate is calculated using the following formula:

$$\text{Heart Rate} = \frac{1500}{\text{Number of small squares}} \tag{1}$$

In Eq.1, the formula for this calculation is given. As an example, if the distance between two peaks in the ECG is 25 small squares, then the heart rate would be 60 bpm. Eq.2 shows this fraction (Proven 2019).

$$\text{Heart Rate} = \frac{1500}{25} = 60 \text{ bmp} \tag{2}$$

In the bedside monitoring, the ECG signal is received by the monitor, the heart rate is calculated according to the algorithm the monitor uses, and then it is displayed on the screen.

1.2.2 Respiration Rate

The number of breaths one takes per minute is called respiratory rate. The normal respiration rate for an adult is between 12 to 20 breaths per minute at rest. The respiration rate of under 12 or over 25 is considered abnormal (Grenvik et al. 1972). In the ICU the respiratory rate is usually measured with gas modules connected to the bedside monitoring systems. Figure 3 shows the ECG and Respiration signals.

1.2.3 Blood Pressure

The blood pressure can be obtained using non-invasive or invasive methods. Invasive methods are Arterial Blood Pressure (ABP), Central Venous Pressure (CVP), and Pulmonary Artery Catheter (PAC). The most common ways for measuring Non-Invasive blood pressure (NIBP) is air-filled upper arm cuffs. This approach can be used for measuring both systolic and diastolic pressures (Fortino & Giampà 2010, Li-wei H. et al.

2008).

Blood pressure is recorded as systolic over diastolic blood pressure. Systolic blood pressure is when the heart contracts and pumps the blood. Systolic pressure is the peak pressure. Diastolic pressure is when the heart relaxes and getting filled with blood. Diastolic pressure is the lowest pressure. The unit of measuring blood pressure is mmHg. The normal blood pressure for an adult is $\frac{120}{80}$ mmHg (Tholl et al. 2004, Madell 2018).

Hypertension or high blood pressure puts more stress on the heart and the vessel walls. This increases the possibility of stroke or heart attack. Hypotension or low blood pressure causes dizziness or fainting due to less than needed blood circulation (Madell 2018).

The blood pressure can be measured manually but in the ICU it is done with the NIBP machine. For this, the cuff is first placed on the patient's arm. Then the blood pressure measurement is started by pressing the start button on the machine. The device automatically inflates and deflates. The systolic and diastolic blood pressures are measured and shown on the device's screen.

1.2.4 Pulse Oximetry

Pulse oximetry is measured by a pulse oximeter and it is a non-invasive method for measuring one's oxygen saturation. Oxygen saturation is defined as the fraction of oxygenated hemoglobin relative to total hemoglobin (oxygenated + deoxygenated) in the blood. To function properly the human body requires a very accurate amount of oxygen in the blood. 95–100 percent of oxygen saturation in the blood is the normal amount (Alfred 2020, Sinex 1999).

1.2.5 Temperature

In the ICU the core temperature accurately estimated using the reliable equipment. The patient's core temperature can be measured using invasive or non-invasive methods. In a non-invasive method, a temperature probe is placed on the forehead of the patient that senses the temperature and sends the signal to the patient monitor. Then the temperature is displayed on the monitor's screen (Mazgaoker et al. 2017, Cronin & Wallis 2000).

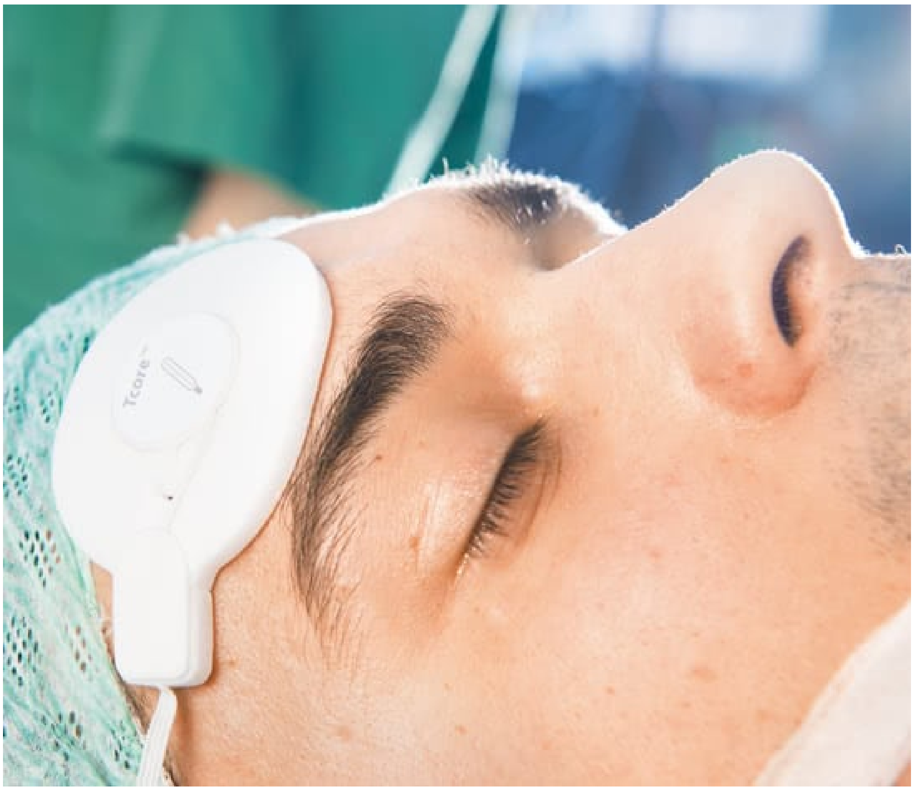


Figure 4. Temperature Probe (Draeger.com 2019)

Figure 4 shows the temperature probe on the patient’s forehead.

2 RELATED WORK

Like in many other areas digitization in healthcare has provided an enormous amount of data. Analyzing and implementing applications based on this data is to generate value as well as improve the clinical care practices. However, the increased number of research in the field of big data in healthcare does not correspond to the number of applications used in clinical care.

Because the focus of this study is the ICU, this review is started by briefing the process of collecting data and data analysis in the ICU. Then some examples of machine learning models used successfully in the ICU are covered. Finally, some of the challenges for these models to be implemented effectively in the ICU are discussed (Carra et al. 2020). The use of EHR in the ICUs has brought big datasets to consist of categorical data, time series, continuous variables, etc. The data is aggregated in the EHR from bedside monitors, doctor notes from their observations, etc. The complexity, variety, and size of these datasets make it difficult for a human to interpret (Carra et al. 2020). However, they represent a new source of knowledge (Angus 2015).

This new source of knowledge can potentially enhance different areas in healthcare in terms of improving prognostication, developing new diagnostic tools, and personalized patient treatment (Obermeyer & Emanuel 2016). The desired outcome would be developing decision support systems that automatically extract the features from EHRs, do the required processing and finally, show the results on the screens in a digestible format (Obermeyer & Emanuel 2016).

In the ICUs the patients are monitored in a continuous manner. This data along with clinical observations is saved in the EHR. Furthermore, there are still many unknowns about critical illnesses characterized with high-degree of uncertainty (Ghassemi et al. 2015) and scarce clinical evidence (Sanchez-Pinto et al. 2018). Therefore there is a need for data-driven insights to help with the critical illnesses (Citerio et al. 2015, Olson et al. 2015).

To benefit from the potentials of data in the ICU, data professionals and clinicians should conduct a close collaboration for the correct use and interpretation of data. To interpret the data correctly, a good understanding of pathophysiological mechanisms and the standard clinical procedures is required. Because the data can be affected by human errors and missing values. Also, the data can be biased by the standard clinical procedures. Therefore a data quality check is necessary before any analysis (Carra et al. 2020). Based on the problem that needs to be solved a variety of different machine learning algorithms can be applied to the data collected from the ICU to identify the patterns in the data. These algorithms are first trained using a part of the data to find the patterns and then apply what they have learned to the new data.

The quality and quantity of the input features play a big role in the performance and accuracy of the method. In a process called "feature engineering" the input features are chosen. They can be selected by domain expertise, statistically, or both. To enhance the algorithm performance and statistical power the most informative input features should be selected (Carra et al. 2020). For choosing the right algorithm for the problem, there are four main questions to be asked:

- What is the target of analysis?
- What is the quality and nature of the data?
- What is the complexity of the problem to be solved?
- What is the amount of available data?

The problem can be classified first in more generic terms such as classification or regression. Under each class there are different algorithms to be chosen depending on the type of input data and complexity of the problem. A more complex algorithm may require more data and be less explainable. The more interpretable algorithms would have higher chances of being accepted in the clinical practices. Therefore selecting less complex algorithms is favorable.

Model validation is the other important aspect. The model should be validated first internally and then externally. The internal validation is during the training phase and it aims to avoid overfitting and enhancing the model performance. Then in external validation, the model is checked with the dataset that the model has not seen before. Here the aim is to assess how well the model can generalize also how clinically valid the results are (Foster et al. 2014).

After the external validation, two other tests must be performed before using the algorithms in the clinical environment

- Comparing the model accuracy against the accuracy of the current “gold standard”. This "gold standard" can be an estimate by clinical experts, a model in use that is developed formerly or a bio-maker (Carra et al. 2020).

- Prospectively validating the model at the bedside in a completely blinded setting. Having large datasets from different clinical centers would be beneficial in this stage. Furthermore, before model implementation other aspects such as epidemiology, the difference in health systems, and how the pattern for practices should be taken into account to gain a realistic model performance (Carra et al. 2020).

The transparent reporting of model performance in clinical use is necessary. The Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD) guidelines provides best reporting practices (Collins et al. 2015). A few factors such as decision curves (Vickers & Elkin 2006), relative utility analysis (Baker 2009) for using the model at various risk thresholds, and calibration to measure the expected and predicted probabilities are important in model performance (Collins et al. 2015).

Various prognostic scores have been developed so far for baseline risk evaluation, outcome prediction, illness severity characterization, benchmarking (Salluh & Soares 2014) and ICU performance evaluation (Collins et al. 2015, Carra et al. 2020). Two of the prognostic scores are SOFA (Vincent et al. 1996) and APACHE II (Knaus et al. 1985). These scores have three main limitations to be used for patient management in the ICU (Carra et al. 2020).

- Calculating these severity scores, there is an assumption that at the baseline, the patient's physiological values are normal and a certain deviation from these normal values will lead the patient to a specific class. However, there are differences in patients' baseline physiology and this is not considered in these scoring methods, which might lead to the patient's misclassification in the ICU (Deliberato et al. 2018).

- There is a need for these scores to be updated periodically according to the changes in the medical practice
- These scores don't consider how the patient situations evolve in the ICU during the stay, they are only calculated based on the data from the few first hours of the patient's stay.

On the contrary, models created based on the EHR data in the ICU can address these limitations because they can include information about the whole stay. They can be easily updated and recalibrated (Carra et al. 2020). ICU beds are limited resources in several countries (Rhodes & Moreno 2012). ICU admissions are among the most expensive phases of the hospital stay. Therefore there has been an interest to bring more productivity to the way the ICU beds are used. In this regard, good admission, readmission, and discharge policies can make an improvement (Rhodes et al. 2012).

Machine learning algorithms have been used for different types of predictions such as readmission and mortality in the ICU. Rojas et al. (2018) developed a gradient-boosted model to predict ICU readmissions using EHR data. The developed model achieved a better performance than Stability and Workload Index for Transfer Score (SWIFT) (Farmer et al. 2006) and the Modified Early Warning Score (MEWS) (Reini et al. 2012).

The analysis of the data collected in the ICU can bring new insights about diseases and patient management. Shillan et al. (2019) found out that from the 258 studies conducted on the use of machine learning techniques in the ICU, 29.8% targeted at predicting the complications, 27.1% aimed at predicting the mortality, 16.7% focused on developing prognostic models and 11.2% were about patient classification (Carra et al. 2020). Since this thesis work focuses on mortality a review of the relevant studies is given in the following.

Calvert et al. (2016) developed an algorithm called AutoTriage to predict medical intensive care unit mortality using the clinical variables in EHR. They used Multiparameter Intelligent Monitoring in Intensive Care (MIMIC)-III dataset (EW 2016) with a cohort size of 9683 patient records. They used eight commonly measured vitals in ICU, heart rate, pH, pulse pressure, respiration rate, blood oxygen saturation, systolic blood pressure, temperature, and white blood cell count. Their AutoTriage 12 h mortality prediction achieved an AUROC value of 0.88 (95% confidence interval 0.86 to 0.88).

Ye et al. (2020) built several machine learning models to predict mortality in critically ill patients with diabetes. They built and ran Logistic regression, Random Forest, Ada Boost, Gradient Boosting, XGBoost, ANN, Majority Voting and yielded the AUROC of 0.82%, 0.86%, 0.84%, 0.83%, 0.87%, 0.86%, 0.87% respectively. Their cohort consisted of 9954 patients with type 1, type 2, secondary and gestational diabetes. They got their data from MIMIC-III and there were 9954 patients in the MIMIC-III with different types of diabetes and their data set had a prevalence of 1164 (11.69%) deaths and 8790 (88.31%) survivals. They also utilized natural language processing algorithms and developed Knowledge-guided Convolutions Neural Networks (CNN) to clinical notes for predicting mortality.

Kong et al. (2020) used the records of 16,688 sepsis patients from the MIMIC-III dataset to predict the in-hospital mortality of sepsis patients in the ICU using machine learning models. In their cohort, there were 2949 (17.7%) patients who passed away during the hospital stay and 13739 (82,3%) patients who survived the hospital stay. They built Lasso, Random Forest, Gradient Boosting, and logistic regression models for mortality prediction and achieved the AUROC of 0.829, 0.829, 0.845, 0.833 respectively. For building their models they used 86 predicting variables consisting of demographics, laboratory tests, and comorbidities.

Almost all of the studies found in this review used EHR data to predict mortality in the ICU. The only study that used biomedical waveforms to predict mortality was done by Haahti (2019) as the first attempt to use long high-frequency signals to predict mortality in the ICU using deep learning models. At the time of this writing, no studies were found that had used a combination of EHR and biomedical waveforms to predict mortality. Therefore this study is the first one to explore this topic. Before finishing this chapter some of the limiting factors in using the AI algorithms in the ICU are discussed.

Even though many machine learning models have been recently made, only a few applications of them are currently used in ICU for clinical practices. Besides, the clinical value of these models has been assessed only by a few randomized clinical trials.

The lack of AI applications in the ICU can be associated with several factors. Machine learning, like any new discipline, needs to gain the trust of clinicians, this can be achieved through transparency, effective reporting, and encouraging replicability (Vollmer et al. 2020, Fenech et al. 2018). Another important factor that associate with gaining the clinician's trust is the model's interpretability, or how the model provides insights about the input features that contribute to the decision-making process mostly, and to what extent. Many algorithms have been developed during the former years to make the models interpretable (Doshi-Velez & Kim 2017).

The model's trustworthiness depends on its interpretability level, which is determined by center-specific policies and patient consultation results. The algorithms should be useful, besides being trustworthy. There have been several models made without solving any meaningful problems clinically (Vollmer et al. 2020). Therefore, researchers and healthcare personnel for delivering useful and clinically effective algorithms are essential.

Another limitation of AI algorithms in clinical use is that they are usually trained on a specific population dataset, and therefore they can not necessarily heterogeneously be a real-world generalization (Vollmer et al. 2020, Fenech et al. 2018). This can cause inaccuracy in applying these algorithms to certain minorities. The factors mentioned above along with ethical and privacy concerns and the economical demand of ICU digitization are some of the limiting factors in the spread of AI applications in ICU.

3 DATA

3.1 Data Description

The data utilized in this thesis work has been collected from 3/2013 to 12/2015 in 5 adult ICUs at the University of California, San Francisco (UCSF) medical center located in Parnassus Heights neighborhood in San Francisco, California, United States. On this campus faculty, staff, students, and others are engaged in patient care, research, and teaching.

The data contains both EHR and Waveforms. EHR data contains 8 tables. These tables are called Encounters, Diagnoses, FlowsheetRowDim, Flowsheetvaluefact, Lab, medication_orders, Patients, and Procedure_order. Each of these tables contains specific information about the patients. The data is fully anonymized so there are no identifying particulars or details from the patients. This dataset is owned by GE Healthcare and it is not publicly available. This dataset was used by Haahti (2019) in her research towards her thesis work.

In the dataset, there are both Patient_ids and Encounter_ids. Once a patient is admitted to the ICU a Patient_id and Encounter_id is created for them. Next time the same patient is admitted a new Encounter_id is created for them. Since, during the data collection period some patients might have visited for care several times, therefore, the number of Encounter_ids are more than the Patient_ids.

The EHR dataset consists of 2320 patients, counted from the number of unique Patient_ids. With some of the patients being admitted more than once the number of encounters is 5650, counted from the number of unique Encounter_ids. Waveforms are not measured or collected for all the patients in the dataset.

For the total number of 2241 patients, the waveforms during their ICU stay exist in the dataset with varying numbers of signal waveforms collected per patient. For 2221 patients both EHR and waveforms are available in the dataset. The signals considered in this study are ECG, IP, and PPG. Table 2 shows the number of records available for the patients.

Table 2. Number of records available for the patients in the dataset

Cohort	number of patients
Electronic Health Record (EHR)	2320
Waveforms and EHR	2221
Waveforms	2241
Electrocardiogram (ECG)	2241
Impedance Pneumogram (IP)	2240
Photoplethysmogram (PPG)	2240

3.2 Construction of Data

In this section first, the information each table provides is explained. Second, the mappings between the tables are discussed. From the tables in the dataset, the Encounters table includes the patient discharge information and it is used for labeling the data. The FlowsheetValuefact table contains all the vitals measured for patients and it is used for extracting the vitals in feature engineering.

3.2.1 Encounters

The Encounters table includes information about the hospital admissions such as admission and discharge dates, reason of admission, Diagnosis-Related Group (DRG) patient consist in, hospital service required, and discharge disposition. Both Patient_id and Encounter_id information is available in this table. The number of unique Patients_ids in this table is 2317. This table consists of 5650 rows and 27 columns.

3.2.2 Diagnoses

The Diagnosis table consists of the diagnosis of the patient. This table has 230980 rows and 35 columns.

3.2.3 FlowsheetRowDim

FlowsheetRowDim table contains the keys for the variables measured from the patients. In this table, each variable has a key in form of a number and a name. For example, the key for PULSE is 38524, the key for RESPIRATIONs is 39413 and the key for BLOOD PRESSURE is 32710. Using these keys the measured values can be extracted for a Patient_id in the timestamp that is measured. The FlowsheetRowDim consists of 5487 rows representing 5487 different variables and 9 columns.

3.2.4 Flowsheetvaluefact

The table Flowsheetvaluefact contains vital measurements by date and time. The number of unique variables in the flowsheetRowKey column is 7280 that indicates not all of the variables are defined in the FlowsheetRowDim table. The number of variables calculated from FlowsheetRowDim is 5487 and most of them are measured in nominal values. Variables include routine Vital Signs such as blood pressure, pulse, respiratory rate, and temperature. Also other variables like cardiac output, cardiac rhythm, Intracranial Pressure (ICP) readings, and systemic vascular resistance. This table also contains information about drugs, laboratory test results, and patient demographics. This table contains 50450780 rows and 11 columns. The number of unique Patient_ids in this table is 2317.

3.2.5 Lab

The Lab table describes laboratory measurements and tests. The number of rows in the LAB table is 2660829 and the number of columns is 28.

3.2.6 medication_orders

The medication_orders table includes information such as the medication order start time, end time, quantities, and doses. This table contains 562814 rows and 32 columns.

3.2.7 Patients

The Patients table contains patients' demographics information such as age, race, gender, marital status, and preferred language. The number of rows in this table is 2320 that indicates the total number of patients in this dataset and the number of columns in this table is 13. The Encounter_ids are not captured in Patients.

3.2.8 Procedure_order

The Procedure_order table has records of the medical procedures. It has information such as the date and time of the procedure and the name of the procedures conducted. This table has a total number of 2767462 rows and 40 columns.

Table 3. The demographics of patients

Demographics	number of patients(%)
Total patients	2317
Gender	
Male	1207(52%)
Female	1110(48%)
Patient_Status	
Alive	2033(88%)
Deceased	284(12%)
Total encounters	5650

Table 4. Age and length of stay

Statistic	Age(years)	length of stay(days)
Mean	63	8.6
Median	65	5.3
Standard_deviation	15.8	11.9

3.3 Patient Demographics

The demographics for all the patients having EHR data present in this study is demonstrated in table 3. The number of patients based on the Patients table is 2320. However, the information of 3 patients is missing from the Encounters table. Because the Encounters table is used to label data, the demographics are calculated using this table.

3.4 Study Cohort

The study cohort is built by considering the patients that the normal vital signs Blood Pressure, Pulse, Respiratory Rate, and Temperature are measured for them and exist in the dataset. As mentioned before each patient could have visited the ICU several times and that is why the number of Encounter_ids is more. In this study when dealing only with EHR data the Encounter_ids are used. There are in total 5650 Encounter_ids in the dataset out of that 284 encounters led to death while in 5366 encounters the patients recovered.

Flowsheetrowvalefact table that is used for extracting the vitals includes the information for only 5621 encounters that can be divided into two classes of dead and alive (284 dead, 5366 alive). Out of these encounters, for eight of them, all the vitals are not recorded. Therefore the cohort is built with 5613 encounters.

When dealing with EHR data, the Encounter_ids are used instead of Patient_ids. The time period from patient admission to the ICU until the time of disposition is the length of stay. If all the vitals assumed in this study are measured for the encounter of that patient, that encounter is included in the cohort otherwise it is omitted.

3.5 Waveforms

The dataset includes waveform for 2241 patients, in addition to EHR data. The number of biomedical signals measured or available in the dataset varies. In this study, the focus is on three non-invasive waveforms of ECG, IP, and PPG that were selected and used by Haahti (2019)

- There are ECG signals measured from multiple leads in the dataset and sampled with the sampling frequency of 240 Hz. However, only the second lead is used in the study done by Haahti (2019) since usually the morphology of P wave, QRS complex, and T waves are best seen in the second lead (Morris et al. 2009).

- IP is measured from the same electrodes as ECG with the sampling frequency of 60 Hz. The IP signal is then upsampled to 240 Hz to be aligned with ECG signal (Haahti 2019).

- PPG is measured with a pulse oximeter with a sampling frequency of 60 Hz. Then the signal is upsampled to 240 Hz to have the same sampling rate as the IP and ECG signals (Haahti 2019).

Form the whole number of patients, 2320, both waveform and EHR data are available for only 2221 patients. ECG, IP, and PPG signals are measured for almost all of these patients who have both EHR data and waveforms.

All the prepossessing on the waveforms are done by Haahti (2019) with detailed information on the discontinuities in the waveform data and the inclusion criteria for building the cohort.

4 RESEARCH METHODOLOGY

In this section, first, the prediction task is defined. Second, window selection for labeling the data is addressed followed by prepossessing the data.

4.1 Prediction Task

The problem is defined as a binary classification one. The target is to predict if the patient in the prediction time window stays alive or dies. Also, this problem is considered as a supervised learning problem and the data is labeled according to the time windows that are detailed in the following sections.

4.2 Time Window Selection

In the first part of the study, using only the EHR data, there are two types of windows selected for doing this prediction task. The first one is called a data extraction window or in short, a data window, and the second one prediction window. The data window starts right after the patient is admitted to the ICU and the measurements are started. The data window is selected to be five hours for this study. The prediction window comes right after the data window, during its time span the situation of the patient is going to be predicted. In this study, the prediction window is selected to be 13 hours.

In the second part of the study for combining the biomedical waveforms and the EHR data, the data window for EHR is set to five hours. During these five hours, one hour of biomedical waveforms data is used that creates a big amount of data. After the data window, there is a one-hour gap follows by 12 hours of prediction window. The change in the size of the prediction window is done to make it possible to elaborate on the work done by Haahti (2019) and being able to compare the results.

The prediction window size selections can be problem-specific or arbitrary. In this study, a fixed size for the prediction window is selected. The data window can be shorter or longer as well as what is set in this study. The window sizes can be considered as a hyperparameter as well to be optimized later. The challenge would be that for each set of window sizes new data labeling and feature engineering should be done. Being able to create a model working with dynamic window sizes can be proposed for the future.

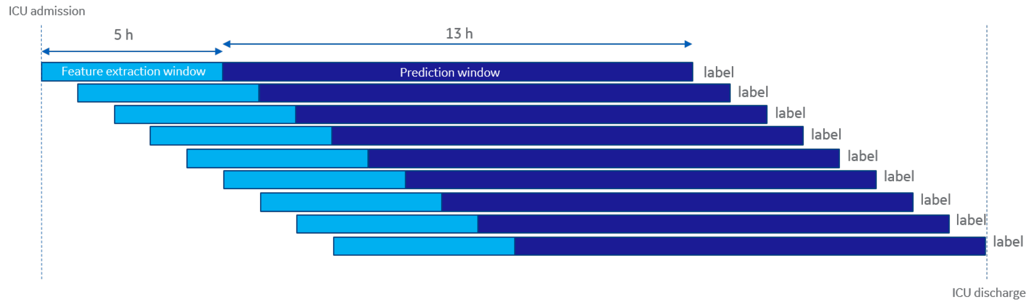


Figure 5. Labeling the EHR data

4.3 Data Prepossessing

The data is located in several different tables. All the values regarding the measurements are in the "value" column of the "Flowsheetvaluefact" table. The patient info and the value for that measurement are depicted in one row of the table. Different values are corresponding to different values available in the table, and they don't have the same measurement frequency.

4.3.1 Making the data ready

First to label the data the Encounters table is used, because this table has a column called "Encounter_Discharge_Disposition" and it contains the information if the patient has deceased or not. Using this info the data can be labeled.

Every patient's stay in the ICU is used to create several instances by using data extraction and prediction windows. As mentioned earlier there are two connected windows, a data extraction window for five hours and a prediction window for 13 hours. These two windows are rolling forward for one hour at a time to create a new instance until the disposition time is reached. For each instance, if the patient has died during the prediction window, that instance is labeled as one otherwise as zero. This is shown in Figure 5. For combining the EHR data and the waveforms the labeling is done similarly. The only difference is that the prediction windows are for twelve hours here and a one-hour gap is set between the data extraction window and the prediction window. This is shown in Figure 6.

After labeling the data, the features are selected. Six vitals, Heart Rate, Respiration, Systolic Blood Pressure, Diastolic Blood Pressure, Pulse Oximetry, and Temperature are

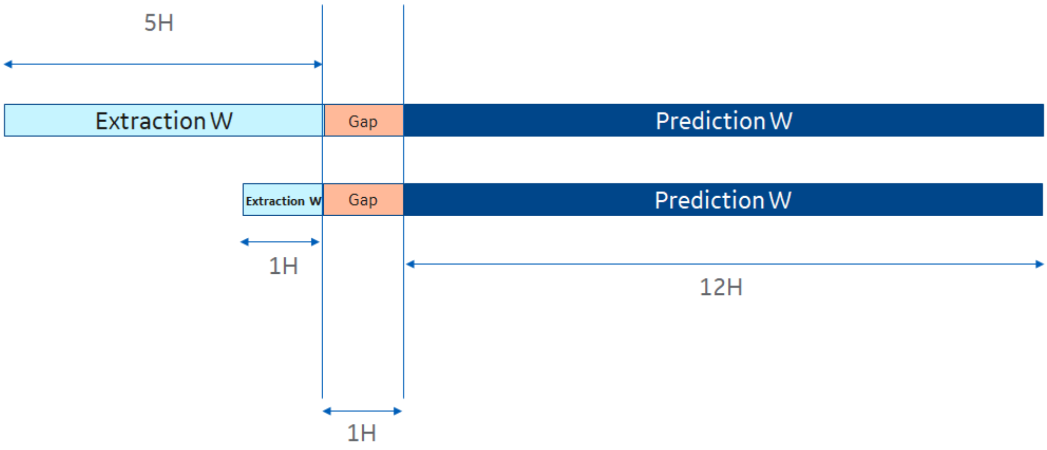


Figure 6. Labeling the EHR data and Waveform

chosen and for each of them, eleven statistics are calculated during the data window. So in total 66 features are gained. Table 7 in the Experiments section shows these statistics.

Then the task in prepossessing is to create a table containing the encounters in the cohort with all the features.

4.3.2 Low/High Range

A static range is used to filter out the vital values that are outside this range. These ranges are shown in table 8 in the experiment section.

4.3.3 Data Scaling

Some of the machine learning algorithms are sensitive to data scaling especially the ones that use gradient descent as the optimization algorithm. To ensure the smooth movement of gradient descent towards the minima with the rate of updated steps for all the features, the data is scaled before being fed to the model. The data can be normalized using Eq.3:

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

for standardization the Eq.4 can be used:

$$X' = \frac{X - \mu}{\sigma} \tag{4}$$

There are also standardization and normalization packages in the scikit-learn library.

4.3.4 Missing Data Imputation

Because of the frequency of measurements for the vitals, during our five-hour data window, some of the values are not available. For filling the missing data forward filling method is used. The forward filling is to fill each missing value with its previous value.

4.4 Class Imbalance

Class imbalance in a binary classification problem means that the number of examples- disease non-disease cases- in the dataset are not the same. This can be also referred to as prevalence or the frequency of the disease.

The class imbalance is common in medical datasets that can be addressed by different methods. One of the methods is to give different weights to each class while building the loss function, more weight to the less frequent class and less weight to the more frequent class to make a balance.

The other method that can be used is resampling. This can be done as undersampling of the examples of more frequent classes or oversampling of the less frequent class to make a balance.

4.5 Metrics

The two main performance metrics commonly used with medical datasets are AUROC and AUPRC. These two metrics are important metrics when it comes to the evaluation of classification problems. Also, they are useful metrics when there is a class imbalance.

Table 5. Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 6. Performance Metrics

Performance metric	Formula
Accuracy	$\frac{TN+TP}{TN+FP+FN+TP}$
True Positive Rate (TPR), sensitivity, recall	$\frac{TP}{FN+TP}$
True Negative Rate (TNR), specificity	$\frac{TP}{TN+FP}$
False Positive Rate (FPR)	$\frac{FP}{TN+FP}$
False Negative Rate (FNR)	$\frac{FN}{FN+TP}$
Positive Predictive Value (PPV), precision	$\frac{TP}{FP+TP}$
Negative Predictive Value (NPV)	$\frac{TN}{TN+FN}$

To be able to understand the AUROC and AUPRC metrics, the confusion matrix should be explained first. Dealing with a binary classification problem, there would be four different outcomes. Prediction of a positive sample correctly will result in a True Positive (TP) and incorrectly will result in False Negative (FN). In the same manner, prediction of a negative sample correctly will result in a True Negative (TN) and incorrectly will result in a False Positive (FP) (Alpaydin 2014).

Table 5 shows a confusion matrix. Based on the confusion matrix several different metrics can be measured. Table 6 shows some of these metrics. Accuracy is a common metric that measures how many samples are classified correctly. However, in medical settings, because medical data sets are usually imbalanced, the measuring accuracy is not as useful.

Sensitivity and recall or True Positive Rate (TPR), measures the proportion of actual positives to all predicted positives. Specificity or True Negative Rate (TNR), measures the proportion of actual negatives to all the predicted negatives. False Positive Rate (FPR) and

False Negative Rate (FNR) can be measured in the same manner as TRP and TNR (Alpaydin 2014).

To classify between two classes a binary classification model requires a threshold. Therefore a fixed threshold is set to calculate the metrics. A change in the threshold means a change in the metrics which makes it difficult to compare the performance of models based on the metrics. The alternative approach for evaluating the performance of models is to use a threshold-free approach. Receiver Operating Characteristic Curve (ROC) and the Precision-Recall Curve (PRC) are examples of this threshold-free approach (Saito & Rehmsmeier 2015).

4.5.1 AUROC

The ROC curve is a plot of the TPR against the FPR at all possible thresholds and shows the trade-off between specificity and sensitivity. The ROC curve of a classifier with random performance is a diagonal line going from (0,0) to (1,1) and can be taken as a baseline. The area under the ROC curve can be measured and shown as a single score that is called Area Under Receiver Operating Characteristic Curve (AUROC). For example, a classifier with random performance has an AUROC value of 0.5 while a perfect classifier would have an AUROC of 1 (Saito & Rehmsmeier 2015).

4.5.2 AUPRC

The PRC curve is a plot of PPV against TPR at all possible thresholds. The area under the PRC curve can be calculated and shown as a single score called Area Under Precision-Recall Curve (AUPRC). The baseline for AUPRC is equal to the fraction of positives (number of positive examples / total number of examples). This means that different classes have different AUPRC baselines. For example, the AUPRC baseline for a class with 20% positives is 0.2, therefore, obtaining an AUPRC of 0.35 for this class is reasonable. On the other hand, a class with 60% positives has an AUPRC baseline of 0.6, so obtaining an AUPRC of 0.30 on this class is not reasonable (Saito & Rehmsmeier 2015).

For balanced data sets, the ROC curve and AUROC are more informative. However, for highly imbalanced datasets, the PRC curve and AUPRC are considered more informative (Saito & Rehmsmeier 2015, Sahiner et al. 2017). In this study, both AUROC and AUPRC values are used to evaluate the models' performance.

4.6 Models

Different models are used in this study. Because the problem is a binary classification, logistic regression (LR) is used as the first model to fit the data and provides a basic understanding of the problem. Furthermore, because explainability and interpretability are important in medical settings the tree-based models are used due to their richness in this aspect.

The tree-based models have shown a high performance with the structured data in the Kaggle competitions. From the tree-based models, Random Forest is used from the bagging pack, and Gradient Boosting (GB) and Extreme Gradient Boosting (XGB) are used from the boosting pack.

For training the models on the waveforms, deep learning models developed by Haahti (2019) are used. Waveforms are considered unstructured data. One hour of waveform data creates a big chunk of data.

Finally for combining the EHR data and waveforms, the predictions from deep learning models are taken and fed to one of the tree-based models as a new feature along with the other features.

4.6.1 Logistic Regression

Since we define our problem as a supervised-learning binary classification to predict if a patient is going to die or recover, the first algorithm to experiment with is Logistic Regression. This algorithm is chosen because of its simplicity and wide usage in supervised-learning binary classification problems. In the following, the logistic regression algorithm is described. In logistic regression, the target or dependent has only two possible classes. That is why it is binary where 1 shows success and 0 failure.

The Logistic Regression can address a multi-class classification problem as well, however, in this writing, only the binary classification is explained. There are several elements in the Logistic Regression algorithm such as hypothesis, cost function, decision boundaries, and gradient descent that need to be understood.

In Logistic Regression the classes are linearly separable and the following formulas are used to calculate the probabilities of samples belonging to each class:

$$p(c=1|\mathbf{x};\mathbf{w},b) = \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} = \hat{y} \tag{5}$$

for the first class, and

$$p(c=o|\mathbf{x};\mathbf{w},b) = 1 - p(c=1|\mathbf{x}) = 1-\hat{y} \tag{6}$$

for the second class, where in Eq. 5 and Eq. 6 **x** is the input vector, **w** represents the weights and b is a bias term. The weights and bias are the parameters for the logistic regression model $z = \mathbf{w}^T x + b$. $c = 1$ denotes the positive class and $\hat{y}$ shows the probabilities for the inputs belong to the positive class. $c = 0$ denotes the negative class and $1-\hat{y}$ shows the probabilities for the inputs belong to the negative class.

Logistic Regression model uses a sigmoid function to map the linear equation $z = w^T x + b$. Eq. 7 shows this sigmoid function:

$$g(z) = \frac{1}{1+e^{-z}} \tag{7}$$

The loss function for the Logistic Regression model can be written as:

$$L(\hat{y}_n, y_n) = -(y_n ln(\hat{y}_n) + (1-y_n) ln(1-\hat{y}_n)) \tag{8}$$

where $\hat{y}_n$ is the model output and y_n is the label.

An average of the losses of each input can be calculated for the approximation error between the label and the predicted output.

$$E(\mathbf{W},b) = \frac{1}{N}\sum_{n=1}^{N}\ln(\hat{y}_n, y_n) = -\frac{1}{N}\sum_{n=1}^{N}(y_n \ln(\hat{y}_n) + (1-y_n)\ln(1-\hat{y}_n)) \tag{9}$$

This total loss function shown in Eq. 9 should be minimized and this is done iteratively with gradient descent, since there is no closed form solution for that. The parameters are initialized randomly and then they are updated step by step as shown in Eq. 10 and Eq. 11. $\nabla_{\mathbf{w}}$ is the partial derivative of $E(\mathbf{W},b)$ in respect of $\mathbf{W}$ and ∇_b is the partial derivative of $E(\mathbf{W},b)$ in respect of b.

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda\nabla_{\mathbf{w}}E(\mathbf{w},b) \tag{10}$$

Eq. 10 shows how **w** is updated, and

$$b \leftarrow b - \lambda\nabla_b E(\mathbf{w},b) \tag{11}$$

Eq. 11 shows how b is updated.

As explained, Logistic Regression is the most widely used model for binary classification problems. However, due to its simplicity, the accuracy gained from training such a model has its limitations. To gain better accuracy different techniques are used.

- **Different algorithms:** In this method, different types of algorithms are trained and the average is taken from them. This method can work well however, each separate algorithm should be built and trained. This will increase the amount of time used on solving the problem dramatically.

- **Different training sets:** In this method, new training sets should be collected and used along with the initial training set. However, this approach might be viable in many cases because it requires access to more data that is not usually available at the desired time.

- **Bagging:** Bagging stands for bootstrap aggregation. Bootstrap is a method in statistics that measures the uncertainty of the estimate. Bagging causes the variance to decrease a bit and bias to increase a bit. But the change in variance is more than the change in bias. Decision trees are high variance and low bias, so bagging is suitable because it decreases the variance for a slight increase in the bias (stanfordonline 2020).

- **Boosting:** Boosting is the other method of ensemble learning in which a bunch of weak learners like decision trees are ensembled for better performance. In this method, each new learner is only trained with the observations that have not been trained effectively by the previous learners. This technique ensures accurate predictions on the vast majority of observations, not only on the easiest ones. So, if an individual model in the team is unable to make good predictions, the other N-1 models will be most likely be able to compensate for it. Boosting is used primarily for reducing bias, and also variance (z_ai 2020c, Breiman 1996).

The machine learning algorithm is called to have a high bias when it can't find the relationship of data points well, or in the other words, the algorithm can't generalize well. That is when underfitting happens.

The machine learning algorithm is called to have a high variance when the accuracy gained from fitting different sets varies. For example, the algorithm fits the training set well but not the test set. A model with high variance and low bias is called an overfitting model.

The desired models are the ones with low bias and low variance. This means the model fits different sets of data well.

4.6.2 Decision Tree

Decision trees are non-parametric models that can be used both for regression and classification. In this study, they are used for classification. This means adding new features does not increase the number of model parameters. Decision trees consist of nodes and branches. Nodes are where the features are evaluated and the split happens. There are three different types of nodes.

1. The root node, this is where the decision tree starts, so this node stays on top and evaluates the feature that splits the data best.

2. Intermediate nodes, these are the nodes used for evaluation of features and they are one layer lower than the root node.

3. Leave nodes, these are where the predictions happen. These are the final nodes in a decision tree.

In each node of a decision tree, the features in the training set are evaluated based on certain metrics, and training set splitting is done using the one feature that provides the best performance. The metrics that are usually used for classification problems are Gini index or Entropy (z_ai 2020a).

Decision trees are easy to use and they have high interpretability. They require fewer data comparing to the other machine learning algorithms and they can tolerate missing data. On the other hand, they can easily overfit the training data. Also, they are weak learners and don't perform well on prediction tasks, that is why they are usually used for creating ensemble models like Random Forest (RF) and Extreme Gradient Boosting (XGB).

4.6.3 Random Forest

Random Forest is an example of bagging ensembling. They are non-parametric models that can be used both for regression and classification. In this study, they are used for classification. Random Forest ensembles many decision trees. Decision trees are considered weak learners. This is done to achieve a better performance than any of the individual learners. The decision trees are easy to explain however they tend to overfit. Therefore they can perform well on the training data but not well on an unseen set of data. Pruning the tree can help with the overfitting problem but on the other hand, can reduce the algorithm predictive power.

Random Forest is a forest of these trees, using the simplicity of each tree and the flexibility of the ensembling method, Random Forests have better performance than decision trees. Also, they are not as susceptible to parameter tuning as decision trees are. However, their interpretability is worse than decision trees. Random Forests are usually built-in 3 phases:

- **Phase 1:** To create a bootstrapped dataset for each of the trees in the forest

  For creating a random forest, there is a need to train N decision trees. For training each of the trees a random sample is collected from the training set. The size of the random sample can be smaller or equal to the size of the training set. In randomizing process one data point can be selected more than once. This process is called bootstrapping that the samples are selected with replacement. This random sampling and training of each tree with a sample reduces the overfittig (z_ai 2020b).

- **Phase 2:** To train the forest with the random datasets we created in phase 1

  Random Forest that is built of decision trees works best if the individual trees are not correlated. So to add more randomness at each node only a subset of all the features are selected and evaluated. Therefore for building each tree two levels of randomizing are used, the first one on the data and the second one on the features. This helps to reduce the variance and achieving a better performing model. Then the same process is done for each of the N decision trees in the random forest (z_ai 2020b).

  1. A bootstrapped data is created for each decision tree.

  2. Each of the bootstrapped data created in the first step is used for creating a decision tree, however, only a subset of features in that data will be used to split on.

  3. These steps would be repeated for making a forest with a variety that determines the excellence of random forest over any single decision tree.

  For using the Random Forest for prediction, we predict with each of the trees and then aggregate the results. In terms of classification, aggregation means finding the mode of prediction. This method of first bootstrapping the data and then aggregating the predictions is called bootstrap aggregation or bagging. As an example, a

Random Forest consisting of 500 decision trees in our classification problem, if 50 of the decision trees predict the patient's death and 450 predicts patient survival, the most frequent prediction, here patient's survival is the Random Forest prediction.

- **Phase 3:** To make predictions with Random Forest

To make predictions with Random Forest we feed each individual tree with the observations for which we like to have a prediction. Then sum up the predictions from each tree to get an aggregated prediction (z_ai 2020b).

4.6.4 Gradient Boosting Models

Boosting works in a way that the N models present in the group are trained sequentially with consideration to the previous model performance on the data. If the previous model did not do a good job on data observations, the weights of that data observation increases. This helps the subsequent models focus on the challenging data observations (z_ai 2020c).

Sometimes boosting models are trained with fixed weights for each learner and instead of assigning an individual weight to each sample, the models are trained to predict the difference of previous predictions on the samples and their real values. This difference is called residual (z_ai 2020c).

Gradient Boosting (GB) Models operate by sequentially training the weak learners, adding more estimators, and predicting the residual errors made by the previous estimators instead of adapting the data weights. Due to this, all the weak models have the same importance. In Gradient Boosting models most of the time, fixed-sized trees are used as base predictors. And these models use a learning rate to reach the results (z_ai 2020c). A comprehensive mathematical explanation of Gradient Boosting is given by Natekin & Knoll (2013) and Friedman (2002).

4.6.5 Extreme Gradient Boosting Models

Similar to Gradient Boosting, in Extreme Gradient Boosting the trees are fit to the residuals of the predictions of the previous trees. The difference is that instead of using fixed-size decision trees used in Gradient Boosting, Extreme Gradient Boosting uses a different kind of trees that are called XGBoost trees. These trees are built by calculating similarity scores for the observations that reached a leave node. XGBoost allows regularisation that can reduce the possibility of individual trees and as a consequence the ensemble model being overfitted. XGBoost models are well optimized to provide the best use of computational resources (z_ai 2020c, Chen & Guestrin 2016)

4.6.6 Deep Learning Models

The deep learning model used in this study is a Convolutional Recurrent Neural Network (CRNN) more specifically a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) model that was developed by Haahti (2019). Figure 7 shows an illustration of this CNN-LSTM model structure that has six convolutional blocks, a Long Short-Term Memory (LSTM) layer, and a dense layer.

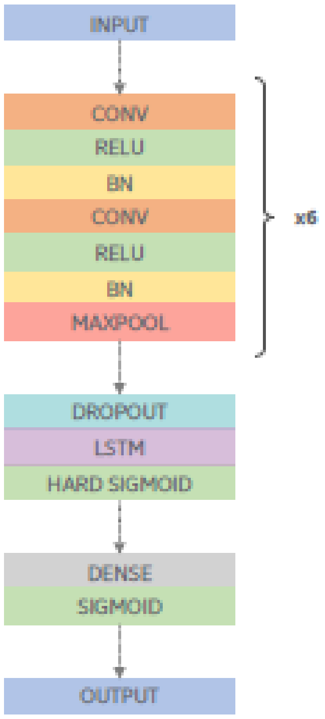


Figure 7. CNN-LSTM model structure (Haahti 2019).

4.7 Baseline Models

In this study Logistic Regression model is taken as the baseline model and the performance of other models is compared to that.

5 EXPERIMENTS

Amazon Web Services (AWS) are used for this study. The data is located in AWS Simple Cloud Storage (S3) and the computations are done on the AWS Elastic Computing (EC2). Amazon Athena is used for getting quick queries by giving Structured Query Language (SQL) commands for the initial data exploration.

Utilizing the EC2 instance, the data is fetched from S3 using Spark Python API (PySpark). Also, the s3fs package makes it easy to read .CSV files into pandas data frames from S3 to EC2 instance and write pandas data frames into .CSV files from EC2 instance to S3.

To be efficient and cost-effective in using the cloud resources first a smaller EC2 instance was created and then based on the need the upgrade was done. In this study, three different types of instances were used. First, t2.micro was used for getting familiar with AWS, then t2.2xlarge was created to train models with the EHR data, and finally g3.8xlarge was used for training the deep learning models with waveform inputs.

The programming language used for prepossessing the data and developing the models is Python. PySpark and other commonly used packages such as SciPy, NumPy, and Pandas are used to prepare the data. For developing the models scikit-learn is used. Also, deep learning models were developed using Keras API, a Python deep learning library, with TensorFlow backend.

As our problem is a supervised learning binary classification, the first task is to label the data. The Encounter_Discharge_Disposition column in the Encounters table contains the categorical data regarding patient status at discharge. One of the categories in this column is if a patient is deceased. This information is used to label the data in the time windows explained in chapter 4.3.1.

Then, data cleaning and prepossessing are done. The vital parameters that are used in the feature engineering part are in the "Flowsheetvaluefact" table "value" column. One of the main tasks in prepossessing is to prepare the data in the format that can be used by the models.

The original data contained both the Patient_id and Encounter_id. Each patient could have been admitted to the ICU multiple times making the number of Encounter_ids more than the Patient_ids. In this study, the Encounter_ids are used when dealing with EHR data. Then for combing the EHR data and waveforms the Patient_ids are used because the work done by Haahti (2019) is based on Patient_ids. Therefore to be able to use the biomedical waveforms and built models in that study a cohort containing the same Patient_ids is made.

As explained in the methodology section the data window for EHR data is set to five hours and for the biomedical waveforms for one hour. The features taken from the EHR data are engineered during these five hours of the data window. The prediction window is set to twelve hours. The duration of the data window and prediction window can be considered problem-specific hence it can be changed. They can also be considered as hyperparameters and be tuned to provide the windows that will result in the best model accuracy.

To extract the features, six separate vitals are considered and for each of them eleven different statistics are calculated during the data window time span. All the features are brought in the table 7. When combining the biomedical waveforms and EHR data, the predictions from the deep learning models trained with biomedical waveforms are considered as an extra feature along with the other features extracted from EHR data.

Table 7. Features extracted from EHR data

Heart Rate	Respiration	Systolic Blood Pressure	Diastolic Blood Pressure	Pulse Oximetry	Temperature
Mean	Mean	Mean	Mean	Mean	Mean
Median	Median	Median	Median	Median	Median
Mode	Mode	Mode	Mode	Mode	Mode
Std Dev	Std Dev	Std Dev	Std Dev	Std Dev	Std Dev
Last Value	Last Value	Last Value	Last Value	Last Value	Last Value
First Value	First Value	First Value	First Value	First Value	First Value
Delta	Delta	Delta	Delta	Delta	Delta
Min	Min	Min	Min	Min	Min
Max	Max	Max	Max	Max	Max
10th %tile	10th %tile	10th %tile	10th %tile	10th %tile	10th %tile
90th %tile	90th %tile	90th %tile	90th %tile	90th %tile	90th %tile

For each vital, an upper and a lower range is used. The ranges are static and they are brought in the table 8.

Table 8. Lower and Upper ranges for the vitals

Vital	Lower range	Higher range
Heart Rate	20	250
Respiration	10	100
Systolic Blood Pressure	10	300
Diastolic Blood Pressure	10	300
Pulse Oximetry	25	100
Temperature	50	111.2

The data is prepared in one table for all the Encounter_ids with all the extracted features. If one of the vitals is not measured for an Encounter_id, that Encounter_id is eliminated from the study. The total number of Encounter_ids in the Encounter table is 5650 (284 dead and 5366 alive), from this amount the vitals are measured for 5621 Encounter_ids in the Flowsheetvaluefact table. In total for 8 Encounter_ids, the vitals are not measured fully so they are removed and the study is done using the 5613 Encounter_ids.

The data is split into a development set and a test set with a ratio of 80/20 percentages. Then the development set is split into train and validation set with the ratio of 80/20. The test data is kept intact. The train validation set is used for training the model and performing the initial evaluations. After that, 5-fold cross-validation is done on the whole development set.

The 80/20 split is done patient-wise to ensure that all data windows from a patient can only be found in one of the sets. This is done to ensure that the model validation is always done with the patients that the model is not trained on. Before data is fed into the models, it is standardized utilizing the StandardScaler library of scikit-learn. Also, performance metrics are calculated using the scikit-learn package.

First Logistic Regression is tried as a simple model for the binary classification task. Then for improving the performance some of the ensemble methods are tried. From the bagging set, Random Forest is used and from the boosting set, Gradient Boosting and Extreme Gradient Boosting are used.

Each model is first trained using a fixed 80 percent of the development set and its performance is measured using the 20 percent of the development set. Various hyper-parameters are used to provide an initial understanding of the models. This is for building a foundation for the model selection to be used in cross-validation.

Then the model is trained with the whole development set using 5-fold cross-validation. To fine-tune the model parameters BayesSearchCV package of scikit-optimize is used. Finally, when the model with the best parameters is achieved, the performance is evaluated upon the test set.

For evaluation of the model performances, the AUROC and AUPRC are used due to an imbalance in the data and that they are commonly used metrics in medical settings. For each run, the performance metrics are computed and a mean, median, and standard deviation for the metrics are captured. Based on these metrics, the models with the best performance selected and retrained with the whole development set. Final models are evaluated with the test set.

To combine the EHR and waveforms a new cohort consisting of 1875 patients, 1500 patients in the development set and 375 patients in the test set is built. A CRNN model built already by Haahti (2019) is trained using 1000 of these patients. Then a prediction is done with the other 500 patients. These predictions are added as a new feature to the 66 features extracted from EHR data. This new dataset is then used for training the Logistic Regression and the ensemble methods.

6 RESULTS

In this section, the results of conducting this study are presented. This section is divided into two parts. In the first part, the results are gained using only EHR data and the best model is identified. In the second part, the results using a combination of EHR data and biomedical waveforms for that best model and Logistic Regression model are demonstrated.

6.1 Cross-Validation Results

In this section, the cross-validation results for all the models are presented. First, only EHR data is used for training the models and the cross-validation results for the four models, Logistic Regression, Random Forest, Gradient Boosting, and Extreme Gradient Boosting are shown. The model with the best performance is selected and then it is trained with the combination of EHR and waveforms along with the Logistic Regression model.

6.1.1 EHR data

Table 9 and Table 10 show the mean, median, standard deviation, minimum and maximum AUROC, and AUPRC performances for the models in the 5-fold cross-validation using the EHR data as the input. Figure 8 visualizes the cross-validation results.

Comparing the AUROC and AUPRC values in the Mean column of tables 9 and 10 show the Extreme Gradient Boosting model achieves the best performance both in terms of AUROC and AUPRC. For example, the XGB models show a performance of 2.3% higher performance in AUROC and 22.2% in AUPRC comparing to the Random Forest which is the second-best model. This increase is 20.27% in terms of AUROC and 175% in terms of AUPRC comparing to the Logistic Regression that is taken as a base model here. Figure 8 visualizes the cross-validation results for the four models.

6.1.2 EHR data and waveforms

In this section, the XGB model as the best performing model and Logistic Regression as the baseline model are trained EHR and waveforms. To do this these two models are first trained with EHR data only and then they are trained with EHR and waveforms.

Table 9. The statistics gained in cross-validation of the models present in this study

AUROC					
Model	Mean	Median	sd	Min	Max
LR	0.74	0.74	0.03	0.70	0.78
RF	0.87	0.86	0.02	0.86	0.91
GB	0.86	0.87	0.02	0.84	0.88
XGB	0.89	0.89	0.02.	0.86	0.92

Table 10. The statistics gained in cross-validation of the models present in this study

AUPRC					
Model	Mean	Median	sd	Min	Max
LR	0.04	0.05	0.01	0.02	0.05
RF	0.09	0.08	0.02	0.07	0.13
GB	0.07	0.06	0.01	0.05	0.09
XGB	0.11	0.12	0.03	0.06.	0.15

Here a new cohort only with a smaller number of patients is used. And the results are not comparable to the last section that all the patients were used to train the models. However, the XGB model still shows the best performance with the cohort. The statistics of other models are not brought to tables for simplicity.

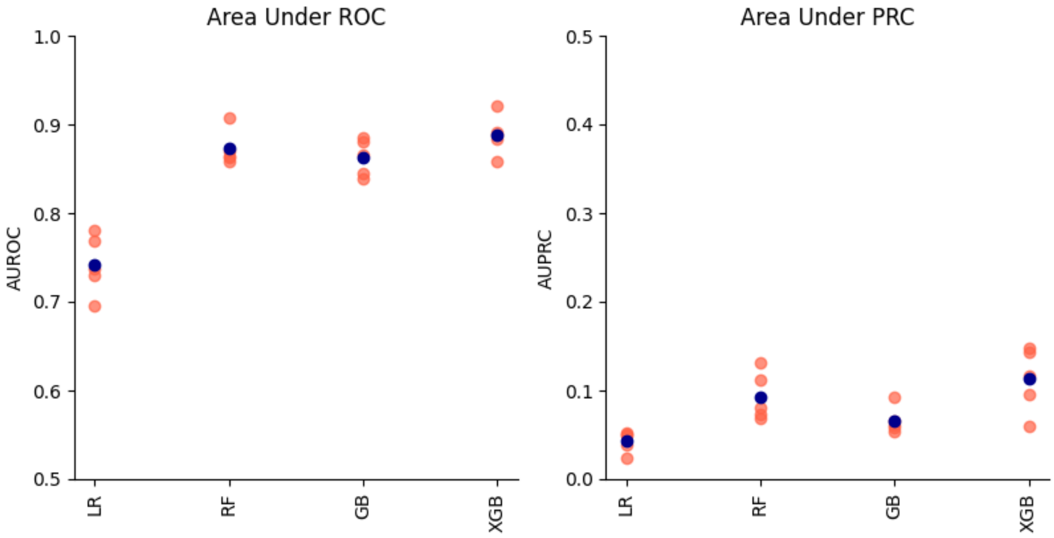


Figure 8. The AUROCs and AUPRCs for the cross-validated LR, RF, GB, and XGB models. The red dots show the individual validation results and the blue dots show the mean value of the cross-validation results for each model.

Tables 11 and 12 show the AUROC and AUPRC for XGB and Logistic Regression models with less number of patients that have all the vitals measurements and the waveforms.

Table 11. The statistics gained in cross-validation of the models present in this study for only EHR

AUROC					
Model	Mean	Median	sd	Min	Max
LR	0.75	0.77	0.10	0.59	0.90
XGB	0.85	0.85	0.09.	0.71	0.97

Table 12. The statistics gained in cross-validation of the models present in this study for only EHR

AUPRC					
Model	Mean	Median	sd	Min	Max
LR	0.11	0.03	0.15	0.02	0.41
XGB	0.27	0.10	0.22	0.07.	0.58

Tables 13 and 14 show the cross-validation results for combining the EHR data and the waveforms as inputs to the models. This combined input improves the performance of the XGB model in training by 1.17% in terms of AUROC and archives a similar performance in terms of AUPRC. Also, the performance improvement for the Logistic Regression model in the training is 5.3% in terms of AUROC and 27% in terms of AUPRC.

6.2 Test Results

In this section, the results gained from evaluating the models with the test set are presented. In the first part, all the models are trained with the whole EHR data and then evaluated with the test set.

In the second part, the model with the best performance and the base model are trained with the new cohort first only with EHR data and then with EHR and waveform data combined. Then their performance is evaluated by the test set.

6.2.1 EHR data

Table 15 summarizes the AUROC and AUPRC scores for the four models evaluated with the test set. These models are trained with the whole training set and then evaluated with

Table 13. The statistics gained in cross-validation of the models present in this study for EHR and waveform

AUROC					
Model	Mean	Median	sd	Min	Max
LR	0.79	0.82	0.11	0.59	0.93
XGB	0.86	0.86	0.09	0.71	0.98

Table 14. The statistics gained in cross-validation of the models present in this study for only EHR and waveform

AUPRC					
Model	Mean	Median	sd	Min	Max
LR	0.14	0.10	0.12	0.04	0.36
XGB	0.27	0.11	0.22	0.08.	0.62

Table 15. The statistics gained in testing of the models present in this study

Model	AUROC	AUPRC
LR	0.736	0.041
RF	0.877	0.095
GB	0.845	0.058
XGB	0.882	0.090

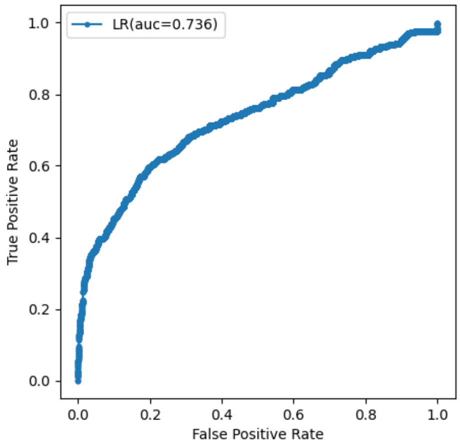
the test set. Figures 9, 10, 11, 12, 13, 14, 15, and 16 show the AUROC and AUPRC for Logistic Regression, Random Forest, Gradient Boosting and Extreme Gradient Boosting models when they are evaluated by the test set.

6.2.2 EHR and waveforms

In this section, the results of evaluating the models with the test set using both EHR and waveform are presented. First XGB and LR models are trained with the EHR data of the new cohort and evaluated with the test set. Then these two models are trained with both EHR and waveform data. Then evaluated with the test set. These two results are compared to see how much improvement combining the EHR data and waveform data provides.

Table 16 summarizes the AUROC and AUPRC scores for the XGB and LR models evaluated with the test set. These models are trained with only the EHR data in the training set of the new cohort and then evaluated with the test set.

Table 17 summarizes the AUROC and AUPRC scores for the XGB and LR models evaluated with the test set. These models are trained with the EHR data plus waveform in the training set of the new cohort and then evaluated with the test set. Figures 17, 18, 19, 20, 21, 22, 23, and 24 show the AUROC and AUPRC for Logistic Regression, and Extreme Gradient Boosting models when they are evaluated against the test set.



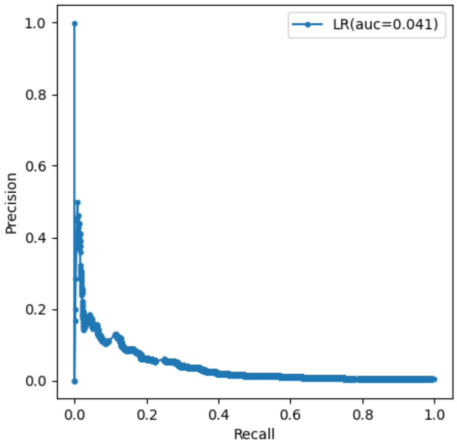
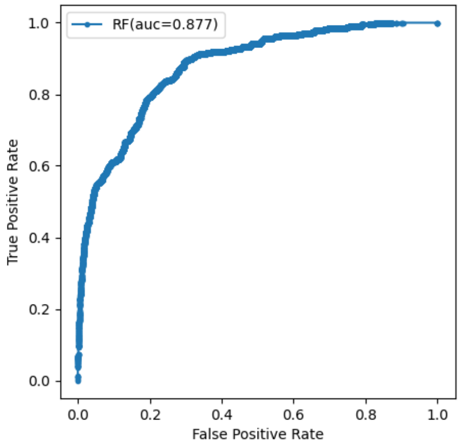


Figure 9. The AUROC for LR using whole EHR

Figure 10. The AUPRCs for LR using whole EHR



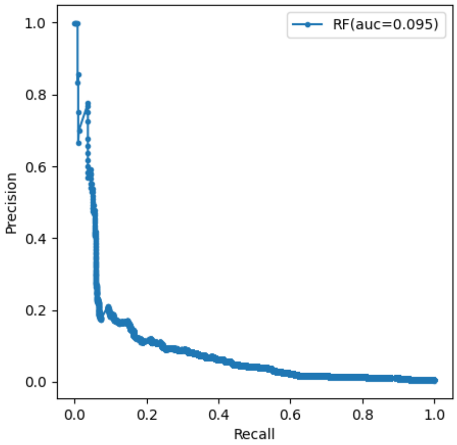
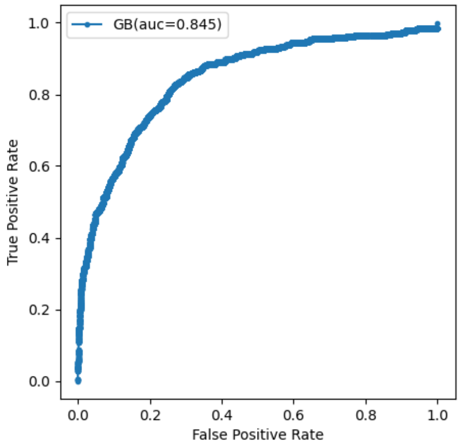


Figure 11. The AUROC for RF using whole EHR

Figure 12. The AUPRCs for RF using whole EHR



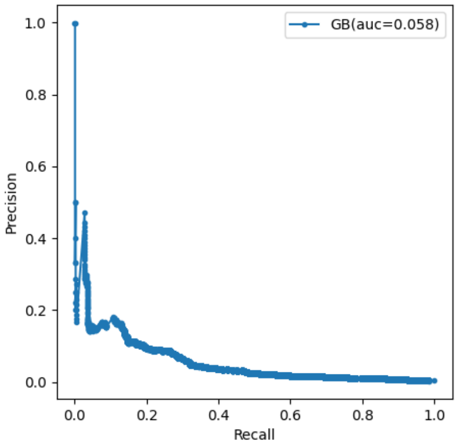
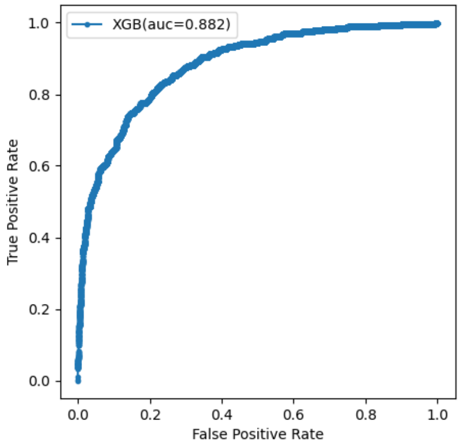


Figure 13. The AUROC for GB using whole EHR

Figure 14. The AUPRCs for GB using whole EHR



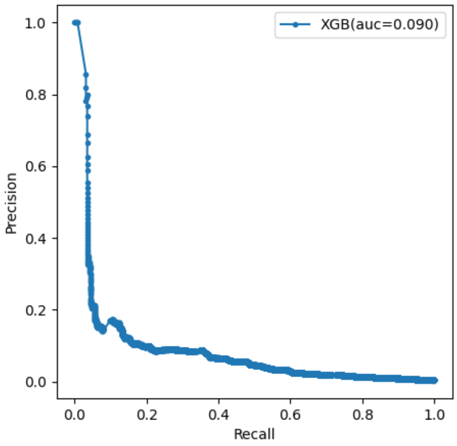


Figure 15. The AUROC for XGB using whole EHR

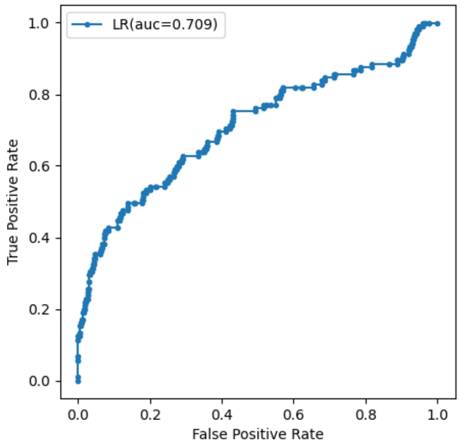
Figure 16. The AUPRCs for XGB using whole EHR

Table 16. Results of evaluation of models with the test set,only EHR

Model	AUROC	AUPRC
XGB	0.851	0.262
LR	0.709	0.189

Table 17. Results of evaluation the models with the test set, EHR + waveforms

Model	AUROC	AUPRC
XGB	0.877	0.289
LR	0.761	0.272



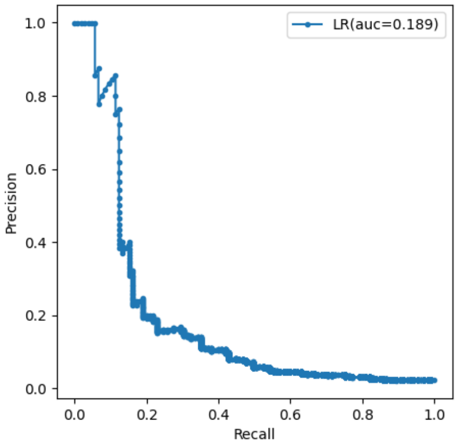


Figure 17. The AUROC for LR, part of EHR

Figure 18. The AUPRC for LR, part of EHR

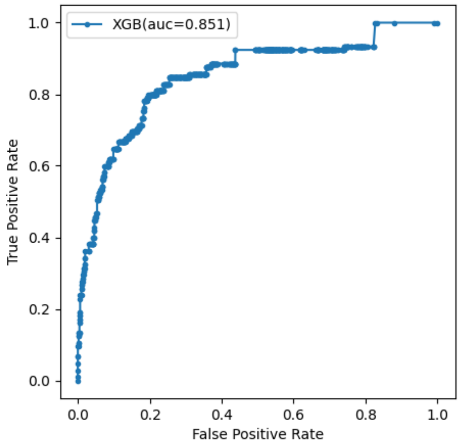


Figure 19. The AUROC for XGB, part of EHR

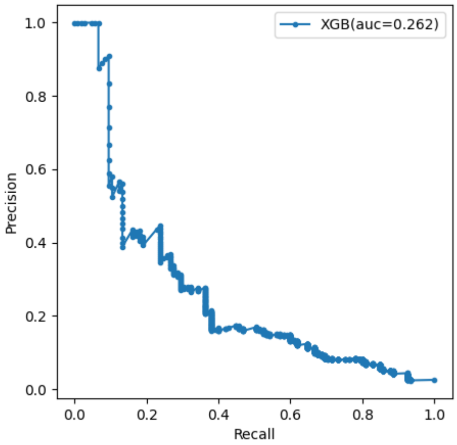


Figure 20. The AUPRC for XGB, part of EHR

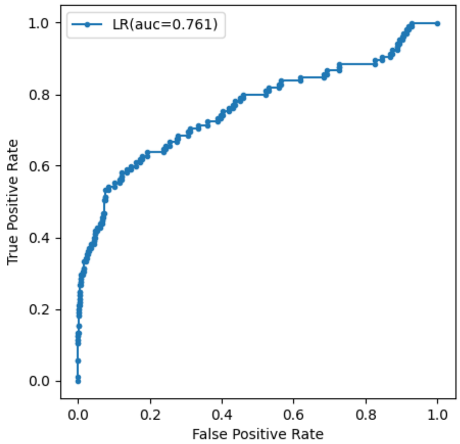


Figure 21. The AUROC for LR, part of EHR+W

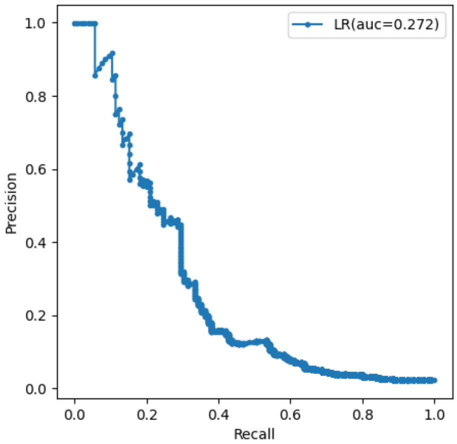


Figure 22. The AUPRC for LR, part of EHR+W

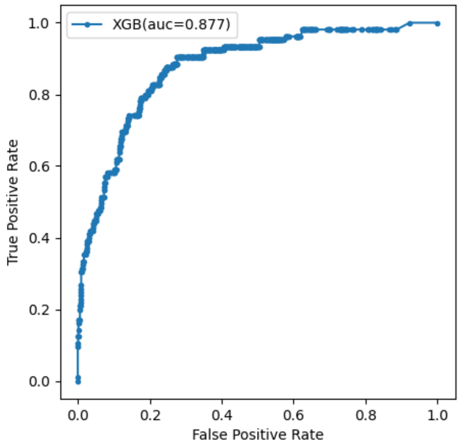


Figure 23. The AUROC for XGB, part of EHR+W

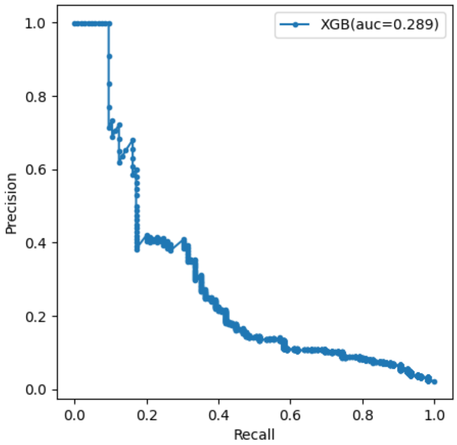


Figure 24. The AUPRC for XGB, part of EHR+W

Combining the EHR and waveform data improves the performance of the XGB model by 3% and the performance of the LR model by 7.3% in terms of AUROC. Also, the performance concerning AUPRC increases by 10.3% for XGB and 44% for the LR model.

7 DISCUSSION

In this study, a machine learning model using the structured data of EHR and unstructured data of biomedical signal waveforms is developed. The objective is to predict mortality in the ICU.

A CRNN model developed by Haahti (2019) is used to do the prediction part based on the large window of waveform data. Then the output of the CRNN model is fed as a feature along with the other features extracted from the EHR to an XGB model. In this section first, the study achievements are described and analyzed. Then, in the second part, some of the limitations and considerations concerning the study settings are introduced.

7.1 Performance of the Proposed Model

This study aims to explore the possibility of combining non-invasive signals and EHR data to predict mortality. First, a model is created using only the EHR data with the whole dataset and the best model reached the performance of 0.88 in AUROC that is the same as the performance achieved by Calvert et al. (2016) using their AutoTriage model, 1.15% higher than the performance achieved by Ye et al. (2020) and 4.15% higher than the performance achieved by Kong et al. (2020) using XGB model. As mentioned in the related work the first two studies used the MIMIC-III dataset with nearly 10000 ICU patients and the third study used the same dataset with more than 16000 patients, however, this study reached the same or higher level of performance with less than 2500 patients in the dataset.

The combination of waveforms and EHR data is tested on the high performer model that is XGBoost and the base model that is the Logistic Regression. The performance of both the high-performance model and the base model is improved when the combination of EHR and waveforms are used as the input compared to using only EHR data or only waveform data.

The CNN-LSTM model developed by Haahti (2019) can achieve an AUROC of 0.84 and an AUPRC of 0.27 using three biomedical signals. The XGBoost model developed in this study that is more explainable and less computationally expensive can achieve AUROC of 0.851 and AUPRC of 0.262 using only EHR data within the same cohort. These two results might not be directly comparable because the two studies are using a different number of vitals, and it is not the goal of this study. Combining the EHR and waveform inputs, the developed model can achieve an AUROC of 0.877 and an AUPRC of 0.289. This is 4.4% higher in terms of AUROC and 7.03% higher in terms of AUPRC compared to the CNN-LSTM model developed by Haahti (2019). Also, this is 3% higher in terms of AUROC and 10.3% higher in terms of AUPRC compared to the similar model fed only with the EHR data.

Haahti (2019)'s study was the first attempt of its kind to use long high-frequency signals with a novel approach, as the previous studies on mortality used only the EHR data or short periods biomedical signals up to 30 seconds. This study as well uses a novel approach to combine biomedical signals and EHR data to improve the performance of models and it is the first study done in this category. This study aims to use all the possibilities of extracting all the information in the waveform data plus EHR and it is significantly different from the existing research, introducing new possibilities in clinical decision-making.

Although in this study combining the biomedical signals and EHR data already improved the performance of the developed model. There is still potential for further exploration in the future for more improvements.

Having a small dataset, 1000 patients to train the deep learning models and 500 to get the predictions is a limitation, and having more data can help the model to generalize better. Also, only one method was used for combing the data however in the future other methods can be used. For example, a multi-modal deep learning model can be used in the future with inputs of EHR and biomedical signals.

7.2 Limitations

Conducting this study, there are some limitations concerning the methods selected and the data. In the following sections, these limitations are discussed.

7.2.1 Data Availability and Quality

Due to varying clinical practices in different hospitals, the studies conducted with the data received only from one site, don't usually show a good generalization when they are tested with datasets from other sites. For example, the data used in this study is from UCSF that is only one site.

Also, data quality plays an important role. For example, the waveforms in this dataset have discontinuities. The amount of noise these discontinuities impose and the effect of them on the results is unknown (Haahti 2019). The models are trained with this dataset containing discontinuities in the waveforms. Therefore, the performance of these models when they are tested with waveforms without discontinuities is unknown.

For the models to be able to generalize better, the data should be collected from different sites in different regions. Besides the dataset is relatively small. Especially when it comes to training the neural network a development set of 1500 patients with a positive prevalence of 3.4% makes it hard for the model to generalize well. In combining the EHR data and biomedical signal waveforms only 1000 patients were used to train the neural network and the other 500 for making predictions to be fed in the next model. This setup makes it even more challenging for the neural network to generalize.

With a larger dataset usually, the model would achieve a better generalization, therefore having more data could improve the results of this study. Also, access to more data makes it possible to use longer data windows and follow the temporary changes in the signals in a longer time span. Longer data windows are also useful when the changes are happening slowly that can't be seen in a short data window.

7.2.2 Prepossessing the Data

The waveform data used as an input to train the deep learning model was preprocessed before being fed to the model. Usually, deep learning models can do the preprocessing in the first layer if they have a large amount of data. Because of the limited number of patients preprocessing was done before feeding the data to the model (Haahti 2019). Furthermore, there are alternatives for each step of preprocessing. For example, imputation could be done with more advanced methods and data normalization could be done differently. That is why the methods used in this study might not be optimal. Although the methods selected for this study are based on previous studies and research, better methods could be found by exploring other possibilities.

7.2.3 Selection of models

Haahti (2019) argues that a combination of CNN and RNN was logical to use to train the waveform data. Because the convolutional and pooling layers make it possible to use a large window of data as input. Besides the recurrent layer makes it possible for the model to learn temporal dependencies of sequences of data. However, there is a huge amount of variation when it comes to the deep learning models. Therefore more research and experimentation might lead to other good approaches.

Also, CRNNs come with a huge number of hyperparameters. Hyperparameter tuning was done to a decent level for CRNNs, LR, RF, GB, and XGB models. However, there might be still room for improvement in this area.

7.2.4 Prediction Task Setting

Prediction tasks usually bring more challenges in comparison to classification tasks. This is due to the uncertainty about the existence of the target in the data that makes the labeling task challenging and questionable. In this study, the discharge time was used as the time of death for labeling the data. However, it is not guaranteed that this time shows the exact time of death.

It is suggested that in the future, the other severe conditions leading to death in case no treatment is given, be considered in the prediction task. However, this is a complicated task with the current dataset because there is limited information about the conditions of patients. In case this approach is taken, labeling should be done with lots of diligence.

The 12-hour prediction window, 5-hour data window, and one-hour gap might not be optimal. These can be considered as hyperparameters in future studies as well. Especially when it comes to the biomedical signal waveforms parts, it is unknown when exactly the death signs appear in the waveform. Therefore, it is not known how far in the future can be predicted.

Haahti (2019) research shows that the model reaches its best performance 8 to 10 hours before the death event. This means that the signs of deterioration are showing themselves several hours before death. This suggests an early detection of the deterioration can be possible.

8 CONCLUSIONS

The study proposed a novel method for exploring the added benefits of combining the EHR data and biomedical waveforms received from non-invasive signals in training machine learning models for predicting mortality in the ICU.

Logistic Regression, Random Forest, Gradient Boosting, and Extreme Gradient Boosting models for fitting the EHR data were implemented and evaluated. The data collection for the model training was done during a five-hour window. Out of these models, XGBoost showed the best performance.

A number of sixty-six features were extracted by using six vital signs of Heart Rate, Respiration, Systolic Blood Pressure, Diastolic Blood Pressure, Pulse Oximetry, and Temperature. For each of these vitals, eleven different statistics were calculated.

A logistic Regression model was considered as the baseline model and the performance of the other models mentioned above was compared to the performance of the base model.

A CRNN based model that was implemented for fitting the waveform data using a one-hour window of waveform data by Haahti (2019) was used in this study to predict the probabilities of patients mortality. These probabilities were used as an additional feature along with the other features to train the XGBoost and Logistic Regression model.

This study was the first attempt to combine long high-frequency signals and EHR data to make longer-term predictions of patient state. The input for the CRNN model consisted of three non-invasive and commonly measured signals (ECG, IP and PPG) for one hour period. The predictions of the CRNN model then were fed to the XGBoost model along with the other sixty-six features.

The models proposed by this study that were fed by both EHR and waveform data outperformed the baseline model by 15.2% in terms of AUROC and by 6.25% in terms of AUPRC. They also outperformed the CRNN models developed by Haahti (2019) by 4.4% in terms of AUROC and 7.03% in terms of AUPRC, as well as the same models when they were fed only with EHR data by 3% in terms of AUROC and by 10.3% in terms of AUPRC.

In this study, one method for combining the EHR and biomedical waveform data was proposed. There are other ways that can be used such as giving both EHR and biomedical waveforms as inputs to a deep learning model. Besides having access to a dataset consisting of more patients, the deep learning models can generalize better. This study proposes these approaches for future work to fully unleash the potentials of combining the biomedical waveforms and EHR data in predicting patient mortality.

REFERENCES

Alfred. 2020, *Pulse Oximetry*, [https://www.youtube.com/watch?v=MHPgamGQmDY]. (Accessed on 04/23/2021).

Alpaydin, Ethem. 2014, *Introduction to Machine Learning, 3rd Editio. ed.*

Angus, Derek C. 2015, Fusing Randomized Trials With Big Data: The Key to Self-learning Health Care Systems?, *JAMA*, vol. 314, no. 8, pp. 767–768. Available: https://doi.org/10.1001/jama.2015.7762.

Aspden, Philip. 2004, Institute of Medicine (US). Committee on Data Standards for Patient Safety.(2004), *Patient safety: Achieving a new standard for care*.

Åstrand, P-O & Ryhming, Irma. 1954, A nomogram for calculation of aerobic capacity (physical fitness) from pulse rate during submaximal work, *Journal of applied physiology*, vol. 7, no. 2, pp. 218–221.

Baker, Stuart G. 2009, Putting risk prediction in perspective: relative utility curves, *JNCI: Journal of the National Cancer Institute*, vol. 101, no. 22, pp. 1538–1542.

Bates, David W; Saria, Suchi; Ohno-Machado, Lucila; Shah, Anand & Escobar, Gabriel. 2014, Big data in health care: using analytics to identify and manage high-risk and high-cost patients, *Health Affairs*, vol. 33, no. 7, pp. 1123–1131.

Baue, Arthur E; Durham, Rodney & Faist, Eugen. 1998, Systemic inflammatory response syndrome (SIRS), multiple organ dysfunction syndrome (MODS), multiple organ failure (MOF): are we winning the battle?, *Shock (Augusta, Ga.)*, vol. 10, no. 2, pp. 79–89.

Bouch, D Christopher & Thompson, Jonathan P. 2008, Severity scoring systems in the critically ill, *Continuing education in anaesthesia, critical care & pain*, vol. 8, no. 5, pp. 181–185.

Breiman, Leo. 1996, *Bias, variance, and arcing classifiers*, Tech. Rep. 460, Statistics Department, University of California, Berkeley ....

Bright, Tiffani J; Wong, Anthony; Dhurjati, Ravi; Bristow, Erin; Bastian, Lori; Coeytaux, Remy R; Samsa, Gregory; Hasselblad, Vic; Williams, John W; Musty, Michael D et al.. 2012, Effect of clinical decision-support systems: a systematic review, *Annals of internal medicine*, vol. 157, no. 1, pp. 29–43.

Bronzino, Joseph D & Peterson, Donald R. 2014, *Biomedical engineering fundamentals*, CRC press.

Cadogan. 2021, *ECG Lead positioning*, [https://litfl.com/ecg-lead-positioning/]. (Accessed on 04/25/2021).

Calvert, Jacob; Mao, Qingqing; Hoffman, Jana L.; Jay, Melissa; Desautels, Thomas; Mohamadlou, Hamid; Chettipally, Uli & Das, Ritankar. 2016, Using electronic health record collected clinical variables to predict medical intensive care unit mortality, *Annals of Medicine and Surgery*, vol. 11, , pp. 52–57. Available: http://dx.doi.org/10.1016/j.amsu.2016.09.002.

Carra, Giorgia; Salluh, Jorge I.F.; da Silva Ramos, Fernando José & Meyfroidt, Geert. 2020, Data-driven ICU management: Using Big Data and algorithms to improve outcomes, *Journal of Critical Care*, vol. 60, , pp. 300–304.

Chen, Tianqi & Guestrin, Carlos. 2016, Xgboost: A scalable tree boosting system, In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Citerio, Giuseppe; Oddo, Mauro & Taccone, Fabio Silvio. 2015, Recommendations for the use of multimodal monitoring in the neurointensive care unit, *Current opinion in critical care*, vol. 21, no. 2, pp. 113–119.

Collins, Gary S.; Reitsma, Johannes B.; Altman, Douglas G. & Moons, Karel G. M. 2015, Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD) : the TRIPOD Statement, *BMC medicine*, vol. 13, no. 1, pp. 1–1.

Cronin, Katrina & Wallis, Marianne. 2000, Temperature taking in the ICU: which route is best?, *Australian Critical Care*, vol. 13, no. 2, pp. 59–64.

Deliberato, Rodrigo Octávio; Ko, Stephanie; Komorowski, Matthieu; Armengol de La Hoz, MA; Frushicheva, Maria P; Raffa, Jesse D; Johnson, Alistair EW; Celi, Leo Anthony & Stone, David J. 2018, Severity of illness scores may misclassify critically ill obese patients, *Critical care medicine*, vol. 46, no. 3, pp. 394–400.

Dick, Richard S; Steen, Elaine B; Detmer, Don E et al.. 1997, *The Computer-Based Patient Record: An Essential Technology for Health Care*, National Academies Press.

Doshi-Velez, Finale & Kim, Been. 2017, Towards a rigorous science of interpretable machine learning, *arXiv preprint arXiv:1702.08608*.

Draeger.com. 2019, *Temperature probe*, [https://www.draeger.com/en$_aunz/Products/Tcore - Temperature - Monitoring - System\#media - gallery]. (Accessedon04/25/2021)$.

EW, Alistair. 2016, Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, *Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. Scientific Data*, vol. 3, , p. 160035.

Farmer, C; Afessa, B; Harris, M; Malinchoc, M; Hubmayr, R & Gajic, O. 2006, Stability and Workload Index for Transfer score predicts unplanned medical ICU patient readmission, *Critical Care*, vol. 10, no. 1, pp. 1–1.

Fenech, M; Strukelj, Nika & Buston, Olly. 2018, Ethical, social, and political challenges of artificial intelligence in health, *London: Wellcome Trust Future Advocacy*.

Fortino, Giancarlo & Giampà, Valerio. 2010, PPG-based methods for non invasive and continuous blood pressure measurement: an overview and development issues in body sensor networks, In: *2010 IEEE International Workshop on Medical Measurements and Applications*, IEEE, pp. 10–13.

Foster, Kenneth R.; Koprowski, Robert & Skufca, Joseph D. 2014, Machine learning, medical diagnosis, and biomedical engineering research - commentary, *BioMedical Engineering Online*, vol. 13, no. 1, pp. 1–9.

Friedman, Jerome H. 2002, Stochastic gradient boosting, *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378.

Gehealthcare.com. 2021, *Patient Monitoring*, [https://www.gehealthcare.com/products/patient-monitoring/patient-monitors]. (Accessed on 04/23/2021).

Ghassemi, Marzyeh; Celi, Leo Anthony & Stone, David J. 2015, State of the art review: The data revolution in critical care, *Critical Care*, vol. 19, no. 1.

Grenvik, Ake; Ballou, Stanley; McGinley, Edward; Millen, J Eugene; Cooley, Wils L & Safar, Peter. 1972, Impedance pneumography: comparison between chest impedance changes and respiratory volumes in 11 healthy volunteers, *Chest*, vol. 62, no. 4, pp. 439–443.

Haahti, Joanna. 2019, *Predicting Mortality with Deep Neural Networks Using Non-Invasive Signals of Intensive Care Patients; Kuoleman ennustaminen neuroverkolla käyttäen teho-osasto potilaiden noninvasiivisia signaaleja*, G2 pro gradu, diplomityö, p. 80. Available: http://urn.fi/URN:NBN:fi:aalto-201906234039.

Haddadi, Ahmed; Ledmani, Mohamed; Gainier, Marc; Hubert, Hubert & De Micheaux, P Lafaye. 2014, Comparing the APACHE II, SOFA, LOD, and SAPS II scores in patients who have developed a nosocomial infection, *Bangladesh Critical Care Journal*, vol. 2, no. 1, pp. 4–9.

Institute of Medicine (US). 2003, *Key capabilities of an electronic health record system : letter report*, Washington, D.C.: National Academies Press, 31 p..

Keegan, Mark T; Gajic, Ognjen & Afessa, Bekele. 2011, Severity of illness scoring systems in the intensive care unit, *Critical care medicine*, vol. 39, no. 1, pp. 163–169.

Knaus, William A; Draper, Elizabeth A; Wagner, Douglas P & Zimmerman, Jack E. 1985, APACHE II: a severity of disease classification system., *Critical care medicine*, vol. 13, no. 10, pp. 818–829.

Kong, Guilan; Lin, Ke & Hu, Yonghua. 2020, Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU, *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–10.

Li-wei H., Lehman; Saeed, Mohammed; Talmor, Daniel & Mark, Roger. 2008, Blood Pressure Measurement in the ICU, vol. 42, no. 2, pp. 157–162.

Madell. 2018, *Blood Pressure*, [https://www.healthline.com/health/high-blood-pressure-hypertension/blood-pressure-reading-explained#normal]. (Accessed on 04/25/2021).

Mazgaoker, Savyon; Ketko, Itay; Yanovich, Ran; Heled, Yuval & Epstein, Yoram. 2017, Measuring core body temperature with a non-invasive sensor, *Journal of thermal biology*, vol. 66, , pp. 17–20.

Medicine, I.; Services, B.H.C. & Safety, C.D.S.P. 2003, *Key Capabilities of an Electronic Health Record System: Letter Report*, National Academies Press. Available: https://books.google.se/books?id=AS16DwAAQBAJ.

Morgan, RJM; Williams, F & Wright, MM. 1997, An early warning scoring system for detecting developing critical illness, *Clin Intensive Care*, vol. 8, no. 2, p. 100.

Morris, Francis; Brady, William J & Camm, A John. 2009, *ABC of clinical electrocardiography*, vol. 93, John Wiley & Sons.

Młyńczak, Marcel; Żyliński, Marek; Niewiadomski, Wiktor & Cybulski, Gerard. 2017, Ambulatory Devices Measuring Cardiorespiratory Activity with Motion.

Natekin, Alexey & Knoll, Alois. 2013, Gradient boosting machines, a tutorial, *Frontiers in neurorobotics*, vol. 7, , p. 21.

Nates, Joseph L; Nunnally, Mark; Kleinpell, Ruth; Blosser, Sandralee; Goldner, Jonathan; Birriel, Barbara; Fowler, Clara S; Byrum, Diane; Miles, William Scherer; Bailey, Heatherlee et al.. 2016, ICU admission, discharge, and triage guidelines: a framework to enhance clinical operations, development of institutional policies, and further research, *Critical care medicine*, vol. 44, no. 8, pp. 1553–1602.

Obermeyer, Ziad & Emanuel, Ezekiel J. 2016, Predicting the future—big data, machine learning, and clinical medicine, *The New England journal of medicine*, vol. 375, no. 13, p. 1216.

Olson, DaiWai M; Kofke, W Andrew; O'Phelan, Kristine; Gupta, Puneet K; Figueroa, Stephen A; Smirnakis, Stelios M; Leroux, Peter D; Suarez, Jose I; Investigators, Second Neurocritical Care Research Conference et al.. 2015, Global monitoring in the neurocritical care unit, *Neurocritical care*, vol. 22, no. 3, pp. 337–347.

Proven. 2019, *ECG*, [https://www.youtube.com/watch?v=S3jtQehfZsw]. (Accessed on 04/23/2021).

Randazzo. 2016, *12-Lead ECG Placement*, [https://www.primemedicaltraining.com/12-lead-ecg-placement/]. Accessed: 2021-03-10.

Rapsang, Amy Grace & Shyam, Devajit C. 2014, Scoring systems in the intensive care unit: a compendium, *Indian journal of critical care medicine: peer-reviewed, official publication of Indian Society of Critical Care Medicine*, vol. 18, no. 4, p. 220.

Reini, Kirsi; Fredrikson, Mats & Oscarsson, Anna. 2012, The prognostic value of the Modified Early Warning Score in critically ill patients: a prospective, observational study, *European Journal of Anaesthesiology (EJA)*, vol. 29, no. 3, pp. 152–157.

Rhodes, Andrew & Moreno, Rui Paulo. 2012, Intensive care provision: a global problem, *Revista Brasileira de terapia intensiva*, vol. 24, no. 4, p. 322.

Rhodes, Andrew; Moreno, Rui Paulo; Azoulay, Elie; Capuzzo, M; Chiche, Jean-Daniel; Eddleston, J; Endacott, Ruth; Ferdinande, P; Flaatten, H; Guidet, B et al.. 2012, Prospectively defined indicators to improve the safety and quality of care for critically ill patients: a report from the Task Force on Safety and Quality of the European Society of Intensive Care Medicine (ESICM), *Intensive care medicine*, vol. 38, no. 4, pp. 598–605.

Rojas, Juan C; Carey, Kyle A; Edelson, Dana P; Venable, Laura R; Howell, Michael D & Churpek, Matthew M. 2018, Predicting intensive care unit readmission with machine learning using electronic health record data, *Annals of the American Thoracic Society*, vol. 15, no. 7, pp. 846–853.

Sahiner, Berkman; Chen, Weijie; Pezeshk, Aria & Petrick, Nicholas. 2017, Comparison of two classifiers when the data sets are imbalanced: the power of the area under the precision-recall curve as the figure of merit versus the area under the ROC curve, In: *Medical Imaging 2017: Image Perception, Observer Performance, and Technology Assessment*, vol. 10136, International Society for Optics and Photonics, p. 101360G.

Saito, Takaya & Rehmsmeier, Marc. 2015, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, *PloS one*, vol. 10, no. 3, p. e0118432.

Salluh, Jorge IF & Soares, Márcio. 2014, ICU severity of illness scores: APACHE, SAPS and MPM, *Current opinion in critical care*, vol. 20, no. 5, pp. 557–565.

Sanchez-Pinto, L. Nelson; Luo, Yuan & Churpek, Matthew M. 2018, Big Data and Data Science in Critical Care, *Chest*, vol. 154, no. 5, pp. 1239–1248. Available: https://www.sciencedirect.com/science/article/pii/S0012369218307256.

Shillan, Duncan; Sterne, Jonathan AC; Champneys, Alan & Gibbison, Ben. 2019, Use of machine learning to analyse routinely collected intensive care unit data: a systematic review, *Critical Care*, vol. 23, no. 1, pp. 1–11.

Sinex, James E. 1999, Pulse oximetry: principles and limitations, *The American journal of emergency medicine*, vol. 17, no. 1, pp. 59–66.

stanfordonline. 2020, *Machine Learning*, [https://www.youtube.com/watch?v=wr9gUr-eWdAt=3615s]. (Accessed on 04/28/2021).

Tholl, Ulrich; Forstner, Klaus & Anlauf, Manfred. 2004, Measuring blood pressure: pitfalls and recommendations, *Nephrology Dialysis Transplantation*, vol. 19, no. 4, pp. 766–770. Available: https://doi.org/10.1093/ndt/gfg602.

Vickers, Andrew J & Elkin, Elena B. 2006, Decision curve analysis: a novel method for evaluating prediction models, *Medical Decision Making*, vol. 26, no. 6, pp. 565–574.

Vincent, J-L; Moreno, Rui; Takala, Jukka; Willatts, Sheila; De Mendonça, Arnaldo; Bruining, Hajo; Reinhart, CK; Suter, PeterM & Thijs, Lambertius G. 1996, *The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure*.

Vollmer, Sebastian; Mateen, Bilal A; Bohner, Gergo; Király, Franz J; Ghani, Rayid; Jonsson, Pall; Cumbers, Sarah; Jonas, Adrian; McAllister, Katherine SL; Myles, Puja et al.. 2020, Machine learning and artificial intelligence research for patient benefit: 20 critical questions on transparency, replicability, ethics, and effectiveness, *bmj*, vol. 368, .

Ye, Jiancheng; Yao, Liang; Shen, Jiahong; Janarthanam, Rethavathi & Luo, Yuan. 2020, Predicting mortality in critically ill patients with diabetes using machine learning and clinical notes, *BMC Medical Informatics and Decision Making*, vol. 20, .

z_ai. 2020a, *Decision Trees Explained. Learn everything about Decision Trees | by z_ai | Towards Data Science*, [https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6]. (Accessed on 04/28/2021).

z_ai. 2020b, *Random Forest Explained. Random Forest explained simply: An easy | by z_ai | Towards Data Science*, [https://towardsdatascience.com/random-forest-explained-7eae084f3ebe]. (Accessed on 04/28/2021).

z_ai. 2020c, *What is Boosting in Machine Learning? | by z_ai | Towards Data Science*, [https://towardsdatascience.com/what-is-boosting-in-machine-learning-2244aa196682]. (Accessed on 05/11/2021).