Master Thesis

Double master's degree of Industrial Engineering
and Automatic Control and Robotics

Developing a machine learning based early warning system for code blue and rapid response events for patient health monitoring

REPORT

Author:	Raimon Padrós I Valls
Professor:	Cecilio Angulo Bahón
Supervisor:	Nicholas E. Houstis
	Aaron D. Aguirre
Call:	Fall 2020
Date:	April 23, 2021

ETSEIB

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona - Universitat Politècnica de Catalunya

UPC

Abstract

This work encompasses the master thesis project for both, Master's degree in Industrial Engineering and Master's degree in Automatic Control and Robotics. It has been carried out at the laboratory of Dr. Aaron Aguirre (Aguirre-lab), which is part of the Massachusetts General Hospital (MGH), a teaching hospital from Harvard Medical School.

In recent years, tons of data are being generated as a result of the continuous monitoring of patients. The diversity and richness of this clinical data present a remarkable opportunity to explore machine learning methods to identify patients at high risk for adverse events such as cardiac or respiratory arrest. In this work, a comprehensive collection of inpatient data records for MGH patients who had a code blue, or a cardiac arrest is assembled and curated. To build the database, a pipeline that collects data from multiple data sources is created. The data collected include demographics, laboratory measurements, low and high-frequency vital signs, and telemetry waveforms among other signals. Furthermore, the pipeline can be used to extend the current database or create a new one with completely different research purposes.

Then, using the aforementioned database, two tree-based machine learning models (Random Forest and XGBoost), as well as the traditional Logistic Regression, are used to create an early warning system for code blue and rapid response events based on electronic health records. The MEWS and NEWS early warning systems are used as benchmarks. Moreover, the impact on the model performance of different training and testing set-ups is explored. To identify which are the signals and features with the highest prediction power, three different feature importance techniques are applied: permutation importance, Gini impurity decrease, and SHAP values. The model is build using physiological variables that are commonly collected by the medical staff for floor patients, such as vital signs, demographics, and results from a comprehensive metabolic panel.

Finally, the research study is complemented by analyzing the economic, social, and environmental impacts, as well as the estimated budget of the project.

Acknowledgments

Funding for this work was provided to the laboratory of Dr. Aaron Aguirre by the Controlled Risk Insurance Company/Risk Management Foundation (CRICO) of the Harvard Affiliated Institutions and by the Hassenfeld Award from the Massachusetts General Hospital, Division of Cardiology.

I gratefully acknowledge the invaluable help, support, and guidance from the advisors and principal investigators of this work, Dr. Aaron Aguirre and Dr. Nicholas Houstis, and for providing my co-workers and me the opportunity to carry out this work in a reputable organization as the Massachusetts General Hospital.

Also, I would like to express my gratitude to all the Aguirre-Lab members, Erik Reinertsen, Roger Pallarès, Eric Palanques, Steven Song, and Ridwan Alam; as well as to Dr. Brandon Westover (CDACs' Executive Director) for the contributions made in the clinical data collection and the helpful discussions and intellectual contributions along the way.

Lastly, I want to extend a word of thanks to my family, who gave me support by all possible means (economically and emotionally) while doing this work in a foreign country during the COVID-19 outbreak.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviation	Description
ADT	Admission, Discharge, and Transfer table
AUC	Area Under the Curve
AuROC	Area under the Receiver Operating Characteristic curve
AuPRC	Area under the Precision-Recall curve
AVPU	Alert, Voice, Pain, Unresponsive
ECG	Electrocardiogram
EDW	Electronic Data Warehouse
EHR	Electronic Health Records
HD5	Hierarchical Data format version 5
ICU	Intensive Care Unit
IHCA	In-hospital Cardiac Arrest
IRB	Institutional Review Board
logreg	Logistic Regression
MGB	Mass General Brigham
MGH	Massachusetts General Hospital
MEWS	Modified Early Warning Score
NEWS	National Early Warning Score
PR	Precision-Recall
PHI	Protected Health Information
ROC	Receiver Operating Characteristic
SHAP	SHapley Additive exPlana-tions
XGBoost	eXtreme Gradient descendent Boosting

Symbol	Unit	Description
sbp	$mmHg$	Systolic Blood Pressure
dbp	$mmHg$	Diastolic Blood Pressure
ppi	U	Pulse Pressure Index
rr	bpm	Respiratory Rate (breaths per minute)
hr	bpm	Heart Rate / Pulse (beats per minute)
SpO_2	%	Oxygen Saturation
Ca	$mmol/L$	Calcium
Na	$mmol/L$	Sodium
K	$mmol/L$	Potassium
Cl	$mmol/L$	Chloride
$agap$	$mmol/L$	Anion Gap
CO_2	$mmol/L$	Carbon Dioxide
bun	mg/dL	Blood Urea Nitrogen
alp	U/dL	Alkaline Phosphatase
ast	U/dL	Aspartate Amino Transferase
wbc	$K/\mu L$	White Blood Cell Count
hgb	g/dL	Hemoglobin
plt	$K/\mu L$	Platelet Count

1. Introduction

The institutional review board (Mass General Brigham Human Research Committee) of the Massachusetts General Hospital approved this study with IRB: #2013P001024, A Database to Support Large-Scale Acute Care Research and IRB: #2020P003053, Cardiovascular and critical care research at MGB.

1.1. Clinical Problem

Medical teams have to face essential treatment decisions every day to prevent urgent events, including cardiac arrest, transfer to the intensive care unit (ICU), mechanical ventilation, intubation, etc. With a limited time and staff to make a clinical decision for each patient [1], they are presented in an overwhelming environment with large amounts of complex data (especially in the intensive care unit patients). These factors, combined with the stress and fatigue caused by extended, night, and rotating shifts [2], compromises the detection of patient deterioration as well as the clinical treatment. Furthermore, the data presented contains valuable information that the clinical staff often ignores by oversimplifying the philology and not being able to obtain features of multivariate signals of risk and waveform properties [3, 4].

From all the in-hospital adverse events, there is particular interest in the early detection of Code Blue and Rapid Response alarms. These correspond to events that are typically related to cardiac arrest episodes and require an immediate medical response to save the patient. Both of them are associated with a high mortality rate. In-hospital cardiac arrests (IHCA) are a public health problem, affecting more than 250.000 patients in the United States over a year, and not even 30% survive to discharge [5].

A Code Blue is the alarm that is activated when a patient has had a cardiac arrest (heart attack, prolonged arrhythmia, etc.) or a respiratory arrest and needs immediate resuscitation. Other reasons to activate a Code Blue alarm include a sudden drop in blood pressure or several stroke signs.

Rapid response is the alarm raised to call for an immediate response of the rapid response team because there is a concern about the patient's condition, especially if there is an abrupt change in any of the vitals or present abnormal values. The rapid response goal is to respond before the cardiac arrest, respiratory arrest, or another severe injury happens.

1.2. Existing Early Warning System

Early detection of patient health deterioration is key to reduce mortality for in-hospital patients. In a clinical setting, simple alarms for individual physiological measurements are used to determine if a patient is at risk. Nevertheless, these unsophisticated systems are not precise enough, miss important events, and overwhelm the medical team with too many false alarms, leading to alarm fatigue [6].

The so-called early warning scores are systems a bit more complex, used to determine the degree of illness of a patient instantaneously. These systems are generally based on a set of vital signs of the patient. The most common early warning systems are the Modified Early Warning Score (MEWS), and the National Early Warning Score (NEWS) [7, 8].

Each of these methods proposes a scoring system based on few vital signs values (Tables 1 and 2). The total score is the sum of the score of each input. The action items corresponding to each final score are shown in Table 3. Note that different institutions might modify slightly both, points assignment and decision thresholds.

Score	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Respiratory Rate [bpm]		$\leq$8		9-14	15-20	21-29	$\geq$30
Heart Rate [bpm]		$\leq$40	41-50	51-100	101-110	111-129	$\geq$130
Systolic Blood Pressure [$mm \cdot Hg$]	$\leq$70	71-80	81-100	101-199		$\geq$200	
Temperature [$°C$]		$\leq$35	35.1-36	36.1-38	38.1-38.5	$\geq$38.6	
Level of Consciousness				A	V	P	U

Table 1. Components and Scoring for the MEWS.

Score	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Respiratory Rate [bpm]	$\leq$8		9-11	12-20		21-24	$\geq$25
Heart Rate [bpm]	$\leq$40		41-50	51-90	91-110	111-130	$\geq$131
Systolic Blood Pressure [$mm \cdot Hg$]	$\leq$90	91-100	101-110	111-219			$\geq$220
Temperature [$°C$]	$\leq$35		35.1-36	36.1-38	38.1-39	$\geq$39.1	
Oxygen Saturation [%]	$\leq$91	92-93	94-95	$\geq$96			
Any supplementary oxygen		Yes		No			
Level of Consciousness				A			V, P, or U

Table 2. Components and Scoring for the NEWS.

Total MEWS Score	Actions
≤3	Regular monitoring
≥4	Continuous monitoring, emergency assessment
Total NEWS Score	**Actions**
0	Monitoring every 12 h
1–4	Monitoring every 4–6 h, assess patient
5–6 or 3 in 1 parameter	Monitoring every hour, emergency assessment
≥7	Continuous monitoring, emergency assessment

Table 3. Score action items.

However, these systems still miss important events and at some institutions are not even useful [9]. It is obvious that, with the appropriate set-up, is possible to implement more complex methods using supervised learning techniques (such as logistic regression and decision trees) that would be able to make predictions as good as, or even better, than these two early warning systems.

1.3 State of the Art

In response to such challenges, the community is exploring different techniques to identify patterns and predictors of adverse events in clinical data for hospitalized patients. Furthermore, the diversity and richness of the clinical data present a remarkable opportunity to use machine learning models [10].

The goal of these machine learning methods is to yield insights to the medical team to diagnose, prioritize, and treat patients. Common outcomes for predictive models are all kinds of emergency events such as transfer to the ICU, death, circulatory failure, and the objectives of this work, code blue, and rapid response events [11, 12].

For some of these endpoints, like cardiac arrest, frequently there are not enough data to develop a supervised learning model, so these outcomes are often combined with others to increase the label prevalence. In [13], Churpek et al. uses a combined outcome of death, cardiac arrest, or transfer to the ICU. Furthermore, in this same work, some insights about which machine learning models perform better for clinical deterioration prediction are given.

Kwon et al. [14] computes a deep learning model to predict cardiac arrest using just four vital signs: systolic blood pressure, heart rate, respiratory rate, and body temperature. In [15], a combined outcome of both cardiac arrest and respiratory failure is predicted using basic vitals and demographics. Note that these works might be the ones that use an outcome closest to the one in this project, as

both, cardiac arrest and respiratory failure prediction are the principal causes for triggering a code blue alarm.

Previous studies focus the analysis to assess the usefulness of different types of predictors such as electronic health records features [16], ECG features [17, 18], and alarms [19].

1.4. Knowledge Gaps

Researchers have explored multiple techniques and data sources; nevertheless, it is not clear which type of signals are more useful neither in which time relative to the event the predictive information is found. What is more, they show the potential as predictors for several types of data, but not when these are combined. In this work, some insights into these questions are given.

Finally, most of the models developed in the literature lack external validation and a deployment/test in a real-time environment. The external validation issue is challenging to overcome, given data protection legislation. Deployment of these algorithms in a clinical setting is something that some researchers are tackling in recent years [20], and indeed, will show up more in the future. These two last aspects are not tackled in this work as they will be faced in a future stage of the project.

1.5. Development Methodology

1.5.1. Software Development Process

The research group have worked under the Scrum methodology. Work sprints of two weeks were defined, the phase associated in each sprint is described in Section 6. Project Timeline. At the end of each sprint, a group meeting was held to plan the next sprint, and review and do a retrospective analysis of the last one. Stand-up meetings were held twice a week. To plan each sprint the GitHub dashboard and issues are used. The post-doctoral researcher has worked as Scrum Manager and the principal investigators of the laboratory as Product Owners.

To guarantee that any tool developed by a group member was accessible and ready to use for any other member, all the progress was uploaded regularly in the repository and documented in a GitHub Wiki. To ensure best practices and identify code bugs, a review from one member was required each time that new content was uploaded into the repository, even if it was not related to one's project.

In Figure 1, is depicted the workflow of the research group. Note that Product Backlog was composed of several projects and each member was focusing on one of these projects. One of the projects is the one described in this report: *Developing a machine learning-based early warning system for code blue and rapid response events for patient health monitoring.*

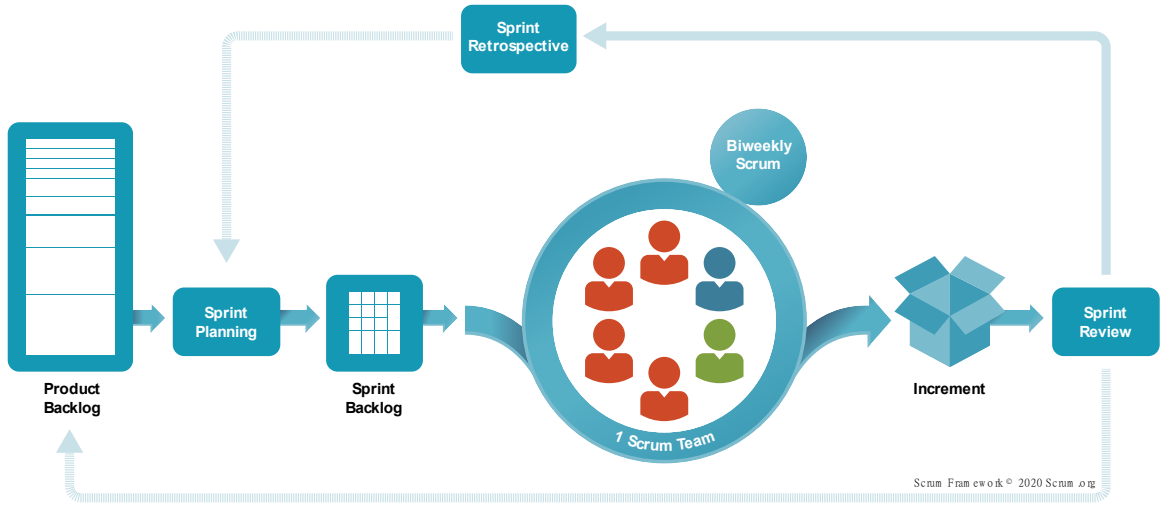


Figure 1. Scrum workflow diagram. Adapted from [21].

1.5.2. Tools

All the study is performed under Ubuntu 20.04.2 LTS platform. Along the study, Structured Query Language (SQL), Python 3.7 (with NumPy [22], Pandas [23], Matplotlib [24], Seaborn[25], h5py [26], Sci-kit learn [27], and XGBoost [28]), and MATLAB® are used. For the manuscript, LaTeX is used. To host and manage the versions of the software developed, GitHub is used.

SQL is a programming language to manipulate, create, and download data from relational databases. It is used to access one of the databases used in the project (EDW).

Python is a high-level programming language. Its extension of powerful libraries makes it a programming language for almost any purpose. Furthermore, it is presented in a readable way that facilitates programming and debugging. It is used to pre-process and manage the data (with NumPy, Pandas, and h5py), create and train the machine learning models (with Sci-kit learn and XGBoost), and create the figures and analyze the results (with Matplotlib, Seaborn, and Sci-kit learn). A brief description of the core packages used is added below:

- NumPy is a scientific computing library for large, multi-dimensional arrays and matrices. Also contains a set of mathematical functions to operate with them.
- Pandas offers structures and methods to operate with tables and time-series data. It is built on top of NumPy.

- h5py is a library to interact with the hdf5 data format.
- Sci-kit learn is a powerful machine learning library containing classification, regression, and clustering models, as well as, a bunch of methods to pre-process the data and analyze the results. It is built on top of NumPy, Matplotlib, Pandas, and other several basic Python libraries.
- XGBoost is a library that extends Sci-kit learn package by adding Extreme Gradient Boosting algorithms. It is built on top of it.
- Matplotlib and Seaborn are plotting libraries for data visualization that have similar features to MATLAB®. Both are built on top of NumPy, and Seaborn on top of Matplotlib.

MATLAB® is a powerful programming and numeric computing platform. In this work it is used to create figures and analyze the results.

LaTeX is a document preparation system. It is especially useful to write scientific documents as it provides tools to cross-reference sections of the document with figures, tables, and references. Furthermore, it helps to keep style consistency along with the document.

GitHub is a git (Global Information Tracker) platform used as a version control system for software development. This platform enables sharing the progress with the members of the research group and work collaboratively in the development of the data collection phase.

2. Objectives, Scope, and Departure Point

2.1. Objectives

The project's main goal is to develop a machine learning model based on Electronic Health Records to predict code blue and rapid response events, with an emphasis on non-ICU floors. This model must trigger an alarm so that it can be used as an early warning system. Furthermore, it has to be based on physiological signals that are commonly collected at any hospital for floor patients, so it is feasible to implement elsewhere. For this purpose, a deep learning model is created, validated, and tested.

Secondly, some of the knowledge gaps stated in the introduction are tackled. More specifically, it aims to answer questions about where the predictive information with respect to the code blue and rapid response alarm time for EHR variables is, as well as which type of features of these signals are more useful.

Bearing this in mind, it is possible to break down the main goal in several specific aims:

1. Assemble a comprehensive collection of inpatient data records for all code blues and rapid responses at the MGH.
2. Coalesce EHR data and continuous ECG telemetry for the study cohort. This database should not be limited to this work's goals but enable different lines of investigation for future research purposes.
3. Build several code blue and rapid response prediction models using different set-ups.
4. Assess the usefulness of the EHR data.
5. Assess the usefulness of the best prediction model as an early warning system.

2.2. Scope

The project encompasses the data collection and the machine learning model implementation and validation using EHR data. Future stages of the project that are not covered in this work, include the study of the use of telemetry and alarms data, the combination of all the different data sources, the implementation in a clinical setting, and the validation with external institution data.

2.3. Departure Point

In the past years, the Aguirre-Lab research group have been exploring the use of machine learning for clinical outcome prediction with external data sources such as The Society and Thoracic Surgeons (STS) [29] database. This project was born out of the idea to extend the lines of investigation using data collected from MGH. This allows external validation of new algorithms within the MGH system and enables the development of new analytics without depending on the availability and access of the data from an external institution.

This report includes the development of the *Developing a machine learning-based early warning system for code blue and rapid response events for patient health monitoring* project from the very early stage of conception and initiation.

3. Methodology

3.1. Clinical Data Collection and Organization

The information intended to use in this work can be found in two different data sources: EDW and Bedmaster. Both databases contain health records for Massachusetts General Hospital patients. To access the databases, one has to be affiliated with the Mass General Brigham association. Furthermore, the Health Insurance Portability and Accountability Act (HIPAA) legislation also requires that the subject is registered under the appropriate IRB and completes specific training about access and management of healthcare data. Therefore, the data collected in this work cannot be shared out of the research team. Protocol #013P001024 grants access to Bedmaster database while Protocol #2020P003053 to EDW.

3.1.1. EDW

The Electronic Data Warehouse (EDW) is a data platform build under SQL Server 2016. It contains all the information collected by Epic (Epic Systems, Verona, Wisconsin) [30] and other databases, such as Commercial, State, and Federal Payers or Patient Financial Data (Strata). In this work, just data derived from Epic is used. Epic is a company that offers software to manage patient care: from registering and scheduling patients' appointments to review and store electronic health records. The EDW data contains Protected Health Information (PHI) and data from all the hospitals associated with the Mass General Brigham health care system. Moreover, any patient that was admitted to the hospital is registered in the EDW database, with a more significant or lower extent amount of data.

The electronic health records collected by Epic, and hence, that are found in EDW contains data formats from numerical values to images, strings, or physician notes. The entire information is sampled at a low-frequency rate (it can vary from minutes to weeks depending on the data type). Every value is measured and approved by medical staff before it is uploaded into the system; for this reason, it is expected that almost no data post-processing will be needed for this data source. The EDW database is updated daily with new patient records.

For each patient, the admission vitals, demographics, diagnoses, clinical events such as alarms, arrests, surgery procedures, vital signs measurements, laboratory values, medical and surgical history, medication information, movements to different hospital floors procedures, and transfusions are collected.

To visualize and obtain EDW data, SQL is used under the Mass General Network. Then, the data is reorganized and filtered using python3 with pandas. The procedure is automatized to obtain all

the information above for a list of patients. The files are stored in .csv format, organized by patient, encounter, and data type.

The EDW data flow from the patient to the server is depicted in Figure 2.

3.1.2. Bedmaster

The Bedmaster data is a high-frequency database that contains information from the bedside monitor. The different devices connected to the bedside monitor collect data from the patient and directly upload it to the server. This data is uploaded deidentified, and it contains just information about the bed and time it was recorded (no PHI). It is possible to identify three different types of data from this source: vital signs, waveforms, and alarms.

The Bedmaster data is usually collected in intensive care unit floors, and with a considerably lower incidence, in the floor beds. So, unlike EDW data, most patients do not have this type of data unless they have been assigned a critical care treatment (patients at risk).

The vital signs and the waveforms are uploaded to the server in their raw format as .stp files (ISO 10303-21). This data sometimes contains time irregularities produced during the recording of the signal by the medical device. To make this database suitable for research, the Clinical Data Animation Center (CDAC) from MGH post-process most of these irregularities and save the data in .mat files (binary data container format from MathWorks - MATLAB®). Each file contains a maximum of 1000 minutes worth of data and corresponds to one or more patients. In other words, the stay of a patient is recorded in multiple files, and the associated recordings are not starting/ending necessarily on the first/last record of a file. Furthermore, there are still time irregularities and overlapping times between files. The .mat files can be found in the MAD3 server, a storage service to provide secure access to large amounts of data from MGH.

Vital signs are sampled every 2 seconds and include signals such as heart rate, respiratory rate, or ventilation rate. Waveforms are sampled at a higher frequency rate: usually at 120 or 240 Hz (depending on the medical device), and include signals such as ECG telemetry, arterial line, and oxygen saturation, among other signals.

Alarms are stored in large .csv files with information such as the alarm name, the bed where it triggered, the time when it triggered, the duration, and the level of severity. The alarms usually trigger when one of the vital signs or waveforms surpass certain thresholds, or a device is failing.

It is decided that this source of data is not going to be used in this stage of the project. First, the extent of patients with such data available is low, with even a lower prevalence in the non-ICU floors.

Secondly, data pre-processing has to be performed differently, which makes it not feasible to perform in the project's duration. Thus, in this work, just the EDW data source is used.

Despite the fact that the data found in this source is not going to be used, it is worth it to coalesce this data together with the EDW data source, so it is possible to broaden the research by adding new data sources without revisiting the data collection phase. In fact, next steps of the project include the use of ECG telemetry data and alarms.

The Bedmaster data flow from the patient to the server is depicted in Figure 2.

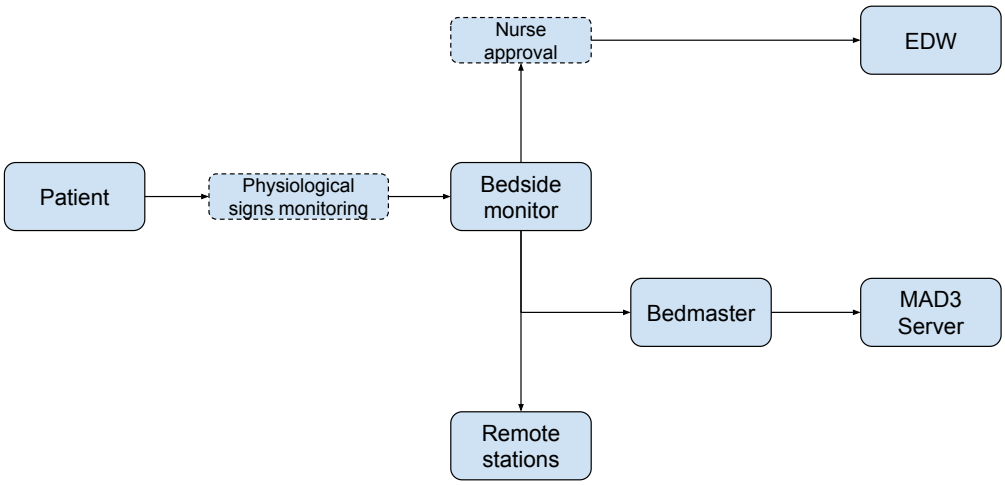


Figure 2. Data collection flow diagram: patient to EDW/Bedmaster.

3.1.3. HD5

To facilitate the feature extraction and the modeling tasks, the data from both data sources have been collected in a unique compressed HD5 file (Hierarchical Data Format) per patient that compounds the cohort. An HD5 file is a container for datasets, which are array-like collections of data, and groups, which are folder-like containers that hold datasets and other groups. Both groups and datasets can contain metadata.

As the data structure and format differ from each data type, each one has been processed differently, but with the same goal: coalesce each patient's signal and save it as a time-series array into the HD5 file. A pipeline that collects and processes all the data for the desired patients has been developed. This pipeline helps to curate the most important data of any cohort of patients, which possibilities different research purposes. The different members of the Aguirre-Lab research group

have collaborated to accomplish that goal. The core contribution of the author of the project has been to pre-process and manage the EDW data source as well as the processing of the Bedmaster alarms. The workflow of this pipeline is depicted in Figure 3.

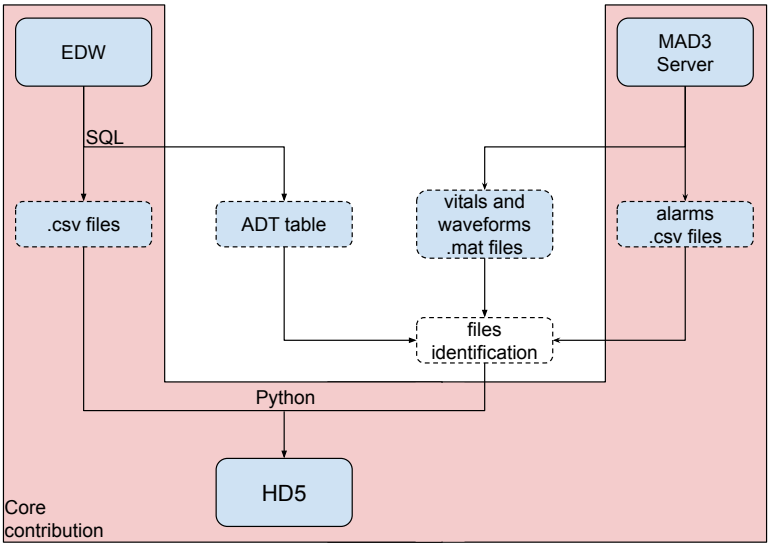


Figure 3. Data collection flow diagram: EDW/Bedmaster to HD5.

The data in each file is organized hierarchically by groups named after the patient encounter, data source, and type of data. The signals available for each patient can be found in the dataset format inside the type of data groups. Additionally, each patient encounter contains metadata describing its demographics and admission related information. The data hierarchy is depicted in the schema in Figure 4.

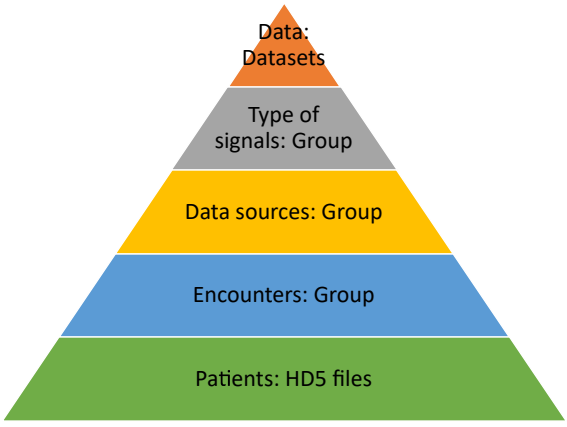


Figure 4. Data hierarchy organization.

For the EDW data source, seven types of signals are found as groups: medications, laboratory measurements, vital signs, surgery procedures, procedures, transfusions, and clinical events. Inside these groups are found the different signals datasets. Invariant information such as units is saved as

metadata.

In order to add the Bedmaster data to the HD5 files, first, it has to be identified. To accomplish such thing, the EDW database is used to pull what is known as an ADT table (Admission, discharge, and transfer). This table contains a time-series of all the records of a patient's movements during their stay at the hospital and the bed and unit it was assigned in each of these movements. Using these records is possible to cross-reference the bed and time information of the Bedmaster data with the corresponding patient.

For the Bedmaster data source, the three different data types explained in 3.1.2 Bedmaster are found as groups: vital signs, waveforms, and alarms. Inside these groups are found the different signals datasets. Invariant information such as units is saved as metadata.

3.2. Prediction Problem

It is hypothesized that tendencies in telemetry data combined with tabular data are strong predictors for emergency events such as code blue and rapid response within the next few hours (prediction window) (4 to 24 hours). As stated in the objectives of this work, just the use of tabular data is explored as predictors.

Being both types of events associated with a high mortality rate, especially the code blue events, there is particular interest in detecting these events the earliest possible and with enough margin time to actuate. Having this idea in mind, a machine learning predictor that defines a probability of having an event in the next few hours will be created, trying to maximize both the sensitivity and the length of the prediction window of the model.

3.2.1. Outcome

The outcome that the model is going to predict is a combination of the occurrence of a code blue or a rapid response in the defined prediction window. As stated in Section 1.1 Clinical Problem, both events correspond to alarm codes that are raised during the stay in the wards of the patients that require an immediate response of the medical team to save the patient. The predictor, whose outcome is the combination of these alarms, will detect the patient's health deterioration, leading to such alarms before the alarm is triggered so the medical staff can actuate on the patient earlier.

The output of the different models will be a probability score between 0 and 1. If the output of the model triggers an alarm will depend on the decision threshold. The decision threshold is chosen corresponding to the 80% sensitivity level. Any score beyond this threshold, an alarm is triggered as the algorithm predicted the patient would have an event within the pre-specified time period. A

score lower than the decision threshold meant that the algorithm is not predicting an event; thus, an alarm is not triggered.

3.2.2. Predictor Variables

Different demographics, vital signs, and routinely collected laboratory values are utilized as predictors. These variables were obtained from the EDW database. Other predictors suggested in the literature, such as alarms or telemetry data, have not been included in this work.

It is important that the model just includes physiological variables that are commonly collected (or at least can be collected at convenience) by the medical staff at any hospital, so it can be implemented in other institutions without major changes. The decision to include each signal is based on the previous statement, the source database's availability, and what the literature shows that have predictive power.

Demographics include age at admission time and time since admission.

Vital signs include heart rate (hr), systolic blood pressure (sbp), diastolic blood pressure (dbp), body temperature ($temperature$), respiratory rate (rr), oxygen saturation (SpO_2), and consciousness level.

Despite that in most of the literature analysis as well as in the MEWS and NEWS warning systems, the AVPU scale is used to assess patient consciousness level, [31] shows that the Glasgow Coma Scale is a more complex but equivalent scale to assess patient consciousness level.

Laboratory values include most of the results of a comprehensive metabolic panel. These include general tests: calcium (Ca), and glucose; electrolytes: sodium (Na), potassium (K), chloride (Cl), anion gap ($agap$), and carbon dioxide (CO_2); kidney function assessment tests: blood urea nitrogen (bun), and creatinine; and liver function assessment tests: bilirubin, albumin, total protein, alkaline phosphatase (alp), and aspartate amino transferase (ast). Furthermore, blood analysis test results are also used: white blood cell count (wbc), haemoglobin (hgb), and platelet count (plt).

Finally, the pulse pressure index (systolic blood pressure minus diastolic blood pressure divided by systolic blood pressure) (ppi) and the bun creatinine ratio ($bun/creatinine$) are computed using the raw data from EDW.

3.2.3. Study Cohort

The observational cohort is composed of all patients from MGH, who were admitted to the hospital between April 2016 (Epic implementation in MGH) and February 2021, and a code blue or a rapid

response alarm was triggered during one of their stays (4344 patients, 2133 code blue, 3258 rapid response). Throughout the manuscript, the time when any of these two alarms is triggered is called event.

For each patient, just the first event is used; it is assumed that the posterior events might contain different physiological information that is not desired to capture in the model created. Patients who had this first event registered less than 48 hours after admission were excluded from the study cohort; this allows the stabilization of patient vitals and the collection of lab samples required by the algorithms. Additionally, events that occurred three months after admission were removed too. Minors (<18 years old) were also excluded from the study cohort. Finally, those patients missing at least one vital sign during the 24 hours period prior to the event were also excluded. After the exclusion criteria, the final cohort comprises 2275 patients and 2275 events, being 498 code blue alarms, and 1777 rapid response alarms. Figure 5 shows a schema of the derivation of the study cohort.

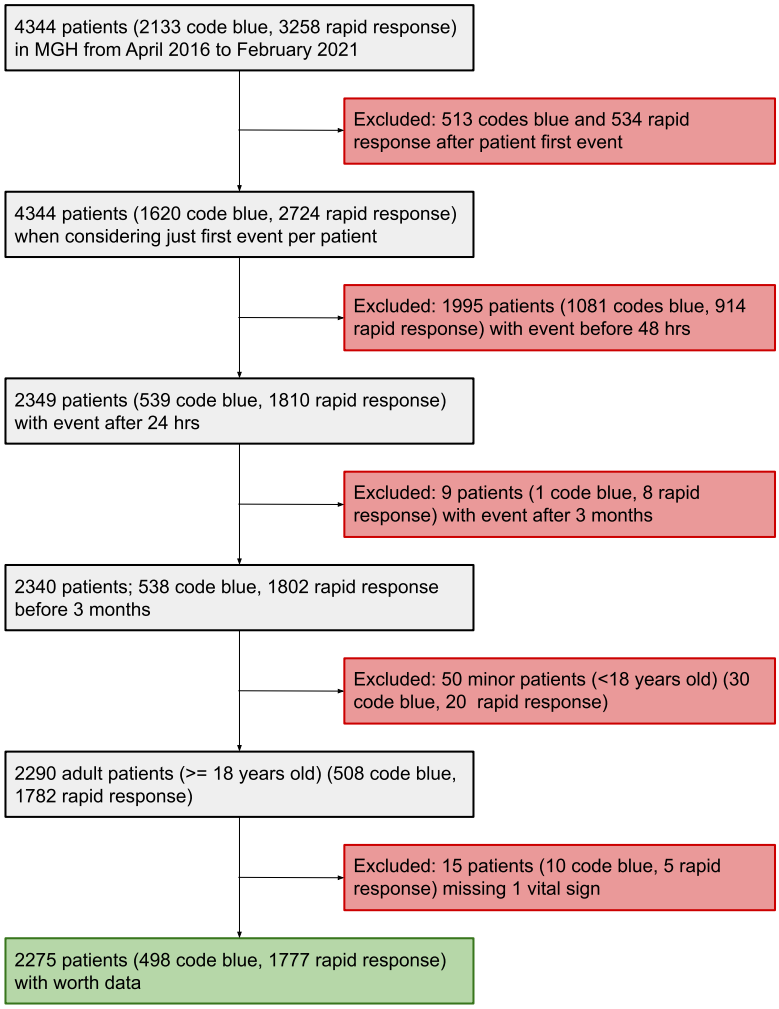


Figure 5. Derivation of the study cohort.

The demographics data for the study cohort are shown in Table 4. Demographics data stratified by code blue and rapid response can be found in Tables A.1 and A.2 respectively from the Appendix A: Supplementary Material section. Note the high mortality rate for these types of alarms, especially for the code blue cohort, is high, which makes the early detection of such events so important. Besides, demographics from both events follow a similar distribution.

n = 2275	**Count (%)**	**Range**	**Mean (Median)**
Sex			
Male	1330 (58.5)		
Female	945 (41.5)		
Alarm type			
Code Blue	498 (21.9)		
Rapid response	1777 (78.1)		
Race			
Black	173 (7.6)		
White or Caucasian	1746 (76.7)		
Other/Unknown	356 (15.6)		
Admission type			
Emergency	2093 (92.0)		
Not emergency	180 (7.9)		
Unknown	2 (0.1)		
Other			
Mortality	1180 (51.9)		
Age (years)		18 - 104	66.38 (68)
Weight (lbs)		57.76 - 494.93	173.82 (166.89)
Height (m)		1.02 - 2.03	1.68 (1.67)
Event time		2 day - 90 days, 5 hrs	10 days, 19 hrs (6 days, 20 hrs)

Table 4. Cohort Demographics.

3.2.4. Problem Definition

The problem is tackled using a Discrete-Time Survival Analysis framework in a binary classification problem. The supervised learning model that is created predicts if a patient will have a cardiac arrest or not within the next few hours.

To obtain a model with such prediction power, it is going to be feed with different samples obtained between the admission and the time of the emergency event of each patient included in the study

cohort. A 48 hours buffer time after admission is used in order to give time for patients' vitals stabilization. This involved separating time into discrete intervals using a fixed step size from forty-eight hours after admission time to the time of the emergency event. In this work, one-hour intervals have been used. For instance, if a patient had an arrest fifty-two hours after admission, the samples will be distributed as shown in Figure 6. Note that this is a simple example, the event usually occurs at a later time of the admission and not in an integer hour.

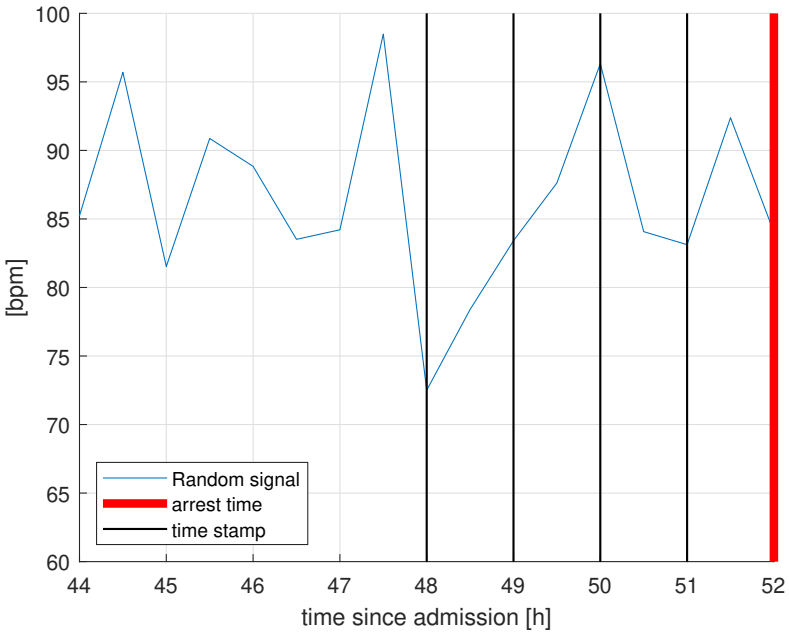


Figure 6. Example of time discretization along a patient's stay.

A sample is defined as positive if the emergency event occurs within a prediction window. The prediction window is defined by a gap time and a prediction horizon with respect to the sample. A gap time of different hours is tested to check if the model is biased towards physiological values that show a patient which is too deteriorated. Furthermore, a model trained using a gap time will predict if an emergency event will occur after the number of hours defined by the gap time, ensuring that the medical team will have at least the amount of time defined by the gap time to respond to the alert. In a similar way, different sets of prediction horizons are used to define how far in advance it is possible to predict these emergency events.

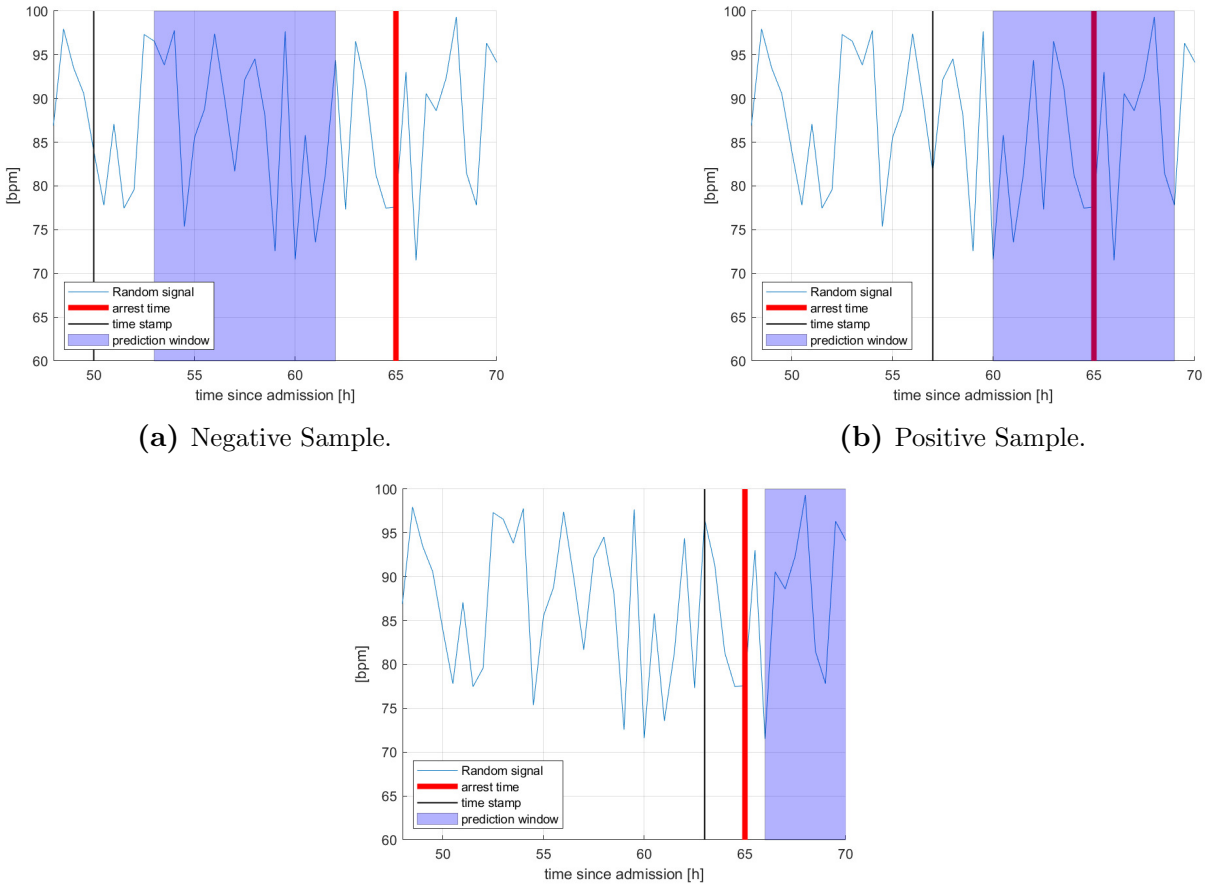
A sample is defined as negative if the arrest occurs after the defined prediction horizon. This means that as control is used data from the same patients who had an emergency event. As the algorithm will be used for patients that might not have an event, random samples from patients without an arrest in their clinical history should be added. The reason why is not implemented in this way is because it requires to define in a different and more complex way the study cohort. It is an issue

that is going to be tackled in the future.

Samples where the event occurs in the gap window are discarded for model training and testing.

The features associated with each sample are computed within a look-back window of different lengths for vital signs and laboratory measurements. For the given window, different metrics of each signal are used as input features of the model. Different lengths of look-back window have been used to define how many hours have worth data for prediction.

Figure 7 shows an example of how samples are labeled as positive, negative, and in the gap window using a prediction horizon of 12 hours and a gap of 3 hours.



(a) Negative Sample.

(b) Positive Sample.

(c) Sample in gap window. Positive for testing, removed from training.

Figure 7. Samples labeling.

Note that using this approach, also known as sliding window, an unbalanced dataset is obtained, as more negative labels are created than positive ones.

During model derivation, 527500 samples where created. The prevalence of an event varies from the time, and gap windows used and vary from 1.5% to 8.5%.

In Table 5 is shown the different set of values used to define both, the look-back window and the prediction window.

Parameter	Values [hrs]
Vital Signs Look-Back Window	[4, 8, 12]
Laboratory Measurements Look-Back Window	[24]
Prediction horizon	[4, 8, 12, 16, 20, 24]
Gap	[0, 1 , 2]

Table 5. Windows parameters.

3.2.5. Dataset Splits

The dataset is split into training, validation, and test sets. An 80% of the data is sampled as training set, 10% is used as validation set to optimize model hyperparameters, and the remaining 10% is held to evaluate the classifier.

To avoid data leak, the split is performed randomly across patients, which means that the multiple samples created with one patient data will contribute only to one of the sets. Note that this means that the % of samples in each split might be different from the 80/10/10 described before, as each patient contributes with a different number of samples.

On top of that, to ensure that each split contains samples of the two different types of events together with different sets of demographics categories, the partition is stratified by event type, mortality, sex, and age. The final training, validation, and test sets are composed of 1820, 227, and 228 patients/events, respectively.

3.3. Data Pre-processing

3.3.1. Features Extraction

A set of 70 features are computed in each window using the different signals mentioned in section 3.2.2. Predictor Variables.

The demographic variables are introduced without transformation. Being the age at admission time a static variable whose value is repeated along the multiple samples of one patient, and the time since admission a variable which is increasing in each window the defined step size (one hour).

For vital signs (except for the Glasgow coma scale), to capture information about the dynamics of the signal along the window, basic metrics are computed. These metrics include the seven following ones: minimum, maximum, mean, first and last value, number of samples in the window (count), and standard deviation. For the Glasgow coma scale, the most recent value is used.

For the laboratory values, as they are sampled less frequently, there is not enough amount of data to perform such statistics, just the last measurement in the window is used. On average, laboratory values are sampled once per day, that is why a fixed look-back window of twenty-four hours for laboratory values is used.

Signal	**min**	**max**	**mean**	**median**	**variance**	**count**
Systolic Blood Pressure [mmHg]	0	300	122,321	119	655,149	267366
Diastolic Blood Pressure [mmHg]	0	182	65,345	63	172,421	267366
Respiration Rate [bpm]	0	198	22,884	20	120,173	270868
Heart Rate [bpm]	0	300	89,34	86	517,174	487828
Pulse Oximetry [%]	0	100	96,325	97	12,302	469181
Body Temperature [°F]	0	140	98,323	98,2	3,372	200115
Glucose [mg/dL]	4	3957	153,021	134	7366,928	41915
Bun [mg/dL]	2	231	37,095	29	739,873	42489
Creatinine [mg/dL]	0,17	18,65	1,763	1,22	2,51	41647
Sodium [mmol/L]	100	180	138,924	139	39,698	41956
Potassium [mmol/L]	1,6	10	4,175	4,1	0,414	42596
Chloride [mmol/L]	51	140	101,534	101	58,046	41530
Carbon dioxide [mmol/L]	2	50	23,678	23	26,84	41324
Anion Gap [mmol/L]	0	59	13,644	13	17,035	41301
Calcium [mmol/L]	1,3	28,4	8,68	8,7	0,661	41633
White blood cell count [K/μL]	0	257,77	11,814	10,07	92,555	36141
Hemoglobin [g/dL]	3,1	21,2	9,385	8,9	4,216	36126
Platelet count [K/μL]	0	1320	197,266	178	17290,709	36133
Bilirubin total [mg/dL]	0,1	61,7	2,456	0,7	30,317	16015
Alkaline phosphatase [U/L]	5	2986	170,221	106	43622,713	15935
Aspartate amino transferase [U/L]	5	33228	161,436	40	754822,288	15910
Albumin [g/dL]	0,4	4790	5,664	2,9	9884,24	16812
Total protein [g/dL]	1,8	11,9	5,943	5,9	1,019	16242
Glasgow Coma Scale	3	15	12,202	14	12,893	53461

Table 6. Summary statistics of the different raw signals between admission and event time for the whole cohort.

In Table 6, the summary statistics across all patients and all time points from admission to arrest time is shown. Note that the vital signs show some values out of a physiologically feasible range. A list of the different combinations of signals and features with the corresponding units can be found in the features table from Appendix A: Supplementary Material, Table A.3. Notice that one of the features listed, *missing indicator*, is introduced later in Section 3.3.3. Missingness.

3.3.2. Outliers

As explained in Section 3.1.1. EDW, all the data that comes from EDW, is verified by medical staff when is introduced into the system. Nevertheless, as the vital signs are introduced manually, there are some values that are misspelled. As seen in the previous section, in Table 6 it can be appreciated that all the vital signs, except for the Glasgow Coma Scale present impossible physiological values (like a body temperature of 0 °F). To remove most of these values, a physiologically feasible range for each vital sign is used. Any sample outside of this range is removed from the dataset. The range limits for each vital sign are shown in Table 7. The range limits are based on the clinical expertise from the doctors of the Aguirre-lab group.

For the laboratory values, as the result is introduced directly by the testing machine and the statistics of the signals shown in Table 6 does not show any abnormal value, no outlier removal technique is applied.

Signal	**Minimum value**	**Maximum Value**
Systolic Blood Pressure [mmHg]	50	280
Diastolic Blood Pressure [mmHg]	20	140
Respiration Rate [bpm]	4	75
Heart Rate [bpm]	10	300
Pulse Oximetry [%]	60	100
Temperature [^{o}F]	85	115
Glasgow Coma Scale	0	15

Table 7. Range outliers for vital signs.

3.3.3. Missingness

For each discrete sample, if a signal was missing in the look-back window, the most recent value for the signal is used (sample-and-hold imputation), no matter how long ago it was taken from the time point of interest. To replace all remaining missing values, mean expected value imputation was performed; mainly at the days after admission or signals that were missing completely. It is assumed

that if a measurement is not present for a patient, it means that there is no reason to think that this value is out of range; thus, it can be represented with an expected healthy value. Then, the computation of a feature x for signal i in time window t is as follows:

- If there is at least one recorded measurement of signal i in window t, all the values in window t are used to compute feature x.
- If there is at least one previously recorded measurement of variable i, forward-filling is performed by setting this previous measurement as a unique measurement value in window t. Then, feature x is computed.
- If there is no previous recorded measurement, the mean expected value is used as a unique value present in the window t. Then, feature x is computed.

When computing metrics for the corresponding signal and window where the value is imputed, the unique imputed value is used to compute the features, and the number of samples in the window (count) is kept to 0. Note that for a given window, if the vital sign value was imputed, metric features $minimum$, $maximum$, $first$, $last$, and $mean$ will have the same value, while features $count$ and $standard\,deviation$ will be set to 0.

It is defined as the expected value for imputation as the mean of the range of healthy values for each signal. The range of healthy values are extracted from the front-end software for clinicians: Hyperspace (Epic). A complete list of the range and the mean healthy value used for imputation for each signal can be found in Table 8. Note that healthy values are distinct from the possible physiological range defined in Table 7.

Literature shows that both missingness and sample frequency can be strong predictors for different outcomes [32]. These types of indicators give information about the attention the medical staff is giving to a patient. It is reasonable to assume that a patient with more frequent laboratory measurements means that the doctor is concerned about this patient's health. Despite an ideal prediction algorithm, it is not desirable that it relies on the medical staff's thoughts and actions; in this stage of the project, it is decided to take advantage of such predictive information. Moreover, it is believed that the number of samples in a window might vary from floor patients to ICU patients.

To do so, the number of samples (count) feature for the vitals is used. This indicator is embedding information of both missingness and sampling frequency. On the other hand, for laboratory values, a missingness flag indicator is added. This indicator is a binary value that defines if the value is present in the window (1) or was imputed in any of the two different forms described above (0). When this feature is added, the models are compound by a total of 88 features.

Signal	Minimum healthy value	Maximum healthy value	Mean healthy value
Systolic Blood Pressure [mmHg]	90	120	105
Diastolic Blood Pressure [mmHg]	60	80	70
Respiration Rate [bpm]	12	18	15
Heart Rate [bpm]	60	100	80
Pulse Oximetry [%]	95	100	97.5
Temperature [^{o}F]	97.7	99.14	98.42
Glucose [mg/dL]	70	100	85
Bun [mg/dL]	6	19	12.5
Creatinine [mg/dL]	0.4	1.2	0.8
Sodium [mmol/L]	135	145	140
Potassium [mmol/L]	3.3	4.5	3.9
Chloride [mmol/L]	98	109	103.5
Carbon dioxide [mmol/L]	24	32	28
Anion Gap [mmol/L]	6	12	9
Calcium [mmol/L]	8.5	10.5	9.5
White blood cell count [K/μL]	4	11	7.5
Hemoglobin [g/dL]	13.5	17.5	15.5
Platelet count [K/μL]	150	450	300
Bilirubin total [mg/dL]	0.2	1.2	0.7
Alkaline phosphatase [U/L]	40	129	84.5
Aspartate amino transferase [U/L]	0	37	18.5
Albumin [g/dL]	3.3	5.2	4.25
Total protein [g/dL]	6	8.5	7.25
Glasgow Coma Scale	15	15	15

Table 8. Range of healthy values for imputation.

Given that missingness and frequency features might bias the model, the impact of including or not including them is studied and analyzed in the results.

3.3.4. Scaling

To better condition the input features, each one is standardized independently by centering the values to zero with unit standard deviation. To do so, first, the mean is removed and then scaled to unit variance as shown in Eq 1. Validation and test sets features are standardized using the mean and standard deviation from training set features. Standardization is preferred towards normalization to not suppress the effect of the extreme values.

Note that some of the supervised learning models defined in Section 3.4. Prediction Algorithms require that the individual features follow a distribution similar to a standard normal distribution.

Otherwise, they would perform poorly.

$$X_{standard} = \frac{X - \mu(X)}{\sigma(X)} \tag{1}$$

3.3.5. Labels

A label of *event within the next hours* or *no event within the next hours* is assigned to each window according to the prediction horizon and gap value used. Then, the categorical labels are encoded in a one-hot numeric array. It is created a binary vector of length 2, where [0, 1] corresponds to the label *event within the next hours* and [1, 0] to *no event within the next hours* as shown in Table 9. The timestamp corresponding to each emergency event was confirmed using the EDW database.

Categorical value	One-hot encoder
Event within the next hours	[0, 1]
No event within the next hours	[1, 0]

Table 9. One-hot encoder for categorical labels.

3.4. Prediction Algorithms

In this work, three different supervised learning models have been tested and compared. These models include a logistic regression [33], and two tree-based methods: a Random Forest [34] and an extreme gradient descendent boosting (XGBoost) [35].

A logistic regression is a linear classification model based on the Sigmoid function (Eq. 2). Where z represents the weighted sum of the features (Eq. 3). The function maps the given value into a probability (value between 0 and 1), and it is assuming that the predictor inputs have a linear relationship with the logit-transformed probability (Eq. 4).

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

$$z = \sum w_i x_i + w_0 \tag{3}$$

$$logit = log_e(\frac{p(z)}{1 - p(z)}) = \sum w_i x_i + w_0 \tag{4}$$

The loss function used by a logistic regression is the Log Loss:

$$Log\ Loss = \frac{1}{n}[\sum_{i=1}^{n} -y_i \cdot log(y_i') - (1 - y_i) \cdot log(1 - y_i')] \tag{5}$$

where n is the number of samples, y_i is the label at sample i, and y'_i the probability predicted by the model at sample i.

To prevent over-fitting, which means a good response in the training data but poor in the test set, two different penalty terms are often added into the aforementioned function: Ridge Regression (L1 regularization) and Lasso Regression (L2 regularization). The Ridge Regression is compound by the squared magnitude of the coefficients (Eq. 6), and the Lasso Regression uses the absolute value of the coefficients (Eq. 7), which shrinks less important features coefficients to zero.

$$Ridge\,Regression\,Penalty = Log\,Loss + \frac{\lambda}{n}\sum_{j=1}^{p} w_j^2 \tag{6}$$

$$Lasso\,Regression\,Penalty = Log\,Loss + \frac{\lambda}{2n}\sum_{j=1}^{p} \mid w_j \mid \tag{7}$$

Where λ is the regularization strength, p the number of features, and w_j the weight of feature j.

When both L1 and L2 regularizations are applied, the model is called elastic-net (Eq. 8), which is the model implemented in this work.

$$loss = Log\,Loss + \frac{\lambda}{n}(l1/l2\sum_{j=1}^{p} w_j^2 + \frac{1}{2}(1 - l1/l2)\sum_{j=1}^{p} \mid w_j \mid) \tag{8}$$

where $l1/l2$ is the regularization ratio.

L1 and L2 regularizers of linear models assume that all features are centered around 0, and variances are in the same order of magnitude. The regularization strength (λ) and the regularization ratio (l1/l2) are the parameters tuned for this model. Logistic Regression models were fit using the Sci-kit learn package for python.

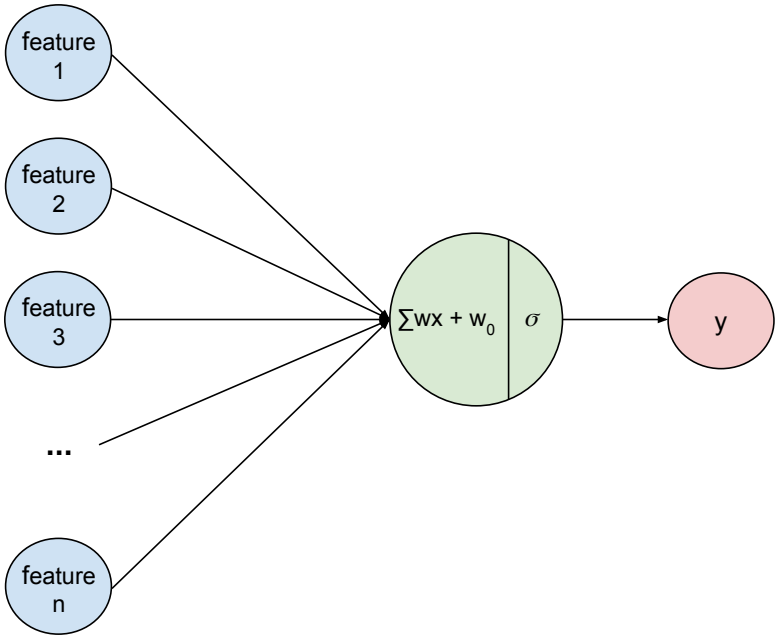


Figure 8. Logistic Regression Model Schema.

A Random Forest is an ensemble of decision trees, which is a series of simple decision rules ("if-then" statements) in the form of a tree with the aim to split the data and arrive to a final classification. A random number of features is considered at each split of each tree. To ensure low correlation among trees, just a subset of the data and features is used to train each tree. The output of all the decision trees is averaged to obtain a unique predictor. It can be seen of a strong predictor integrating a lot of weak predictors

The MEWS and NEWS systems can be represented with a simple decision tree. Thus, it is expected that this model will outperform these alarms systems. The number of trees, the maximum depth of the trees, and the number of features used in each split are the main parameters tuned for this model. Similar to logistic regression, the random forest is easy to understand and interpret by following the decision rules if the number of trees is not too high. Random Forest models were fit using the Sci-kit learn package for python.

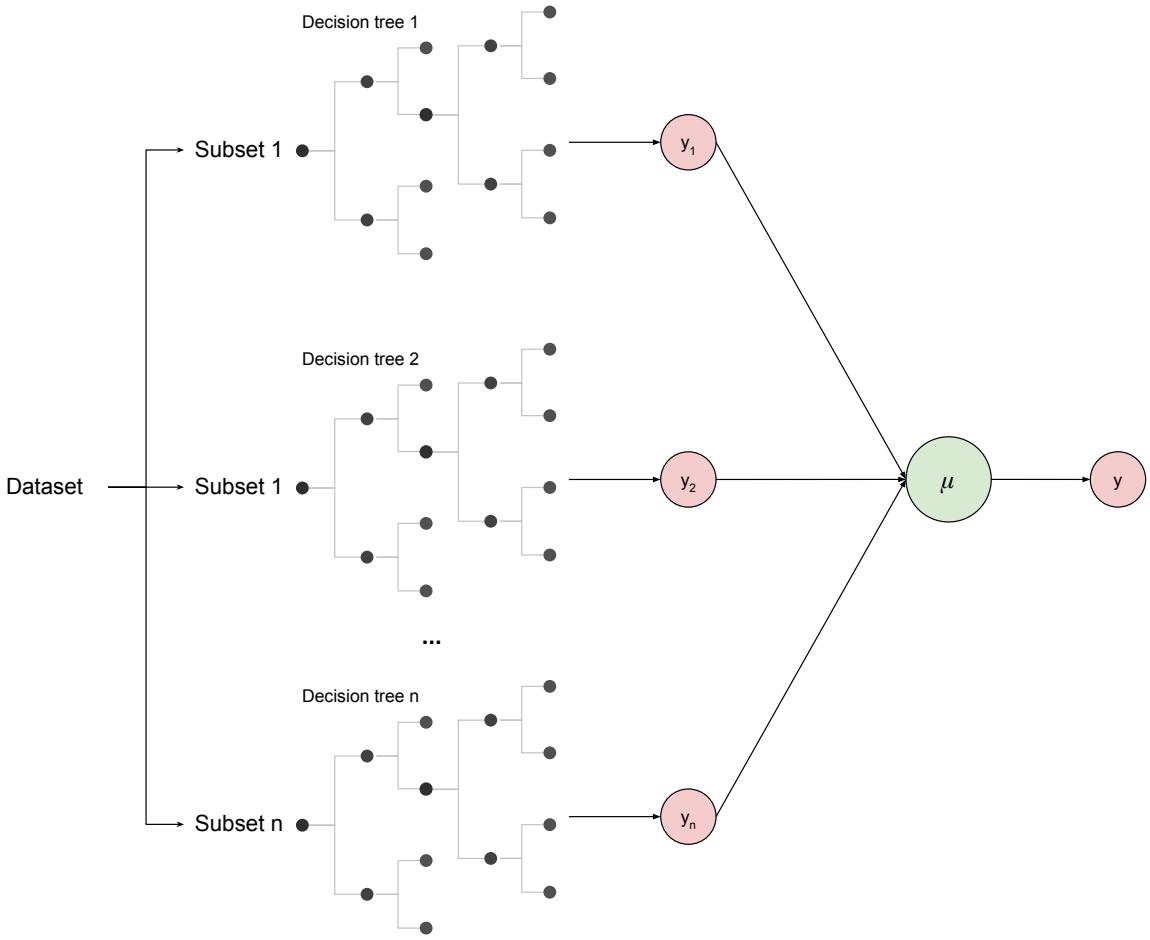


Figure 9. Random Forest Model Schema.

Similar to Random forest, XGBoost is an ensemble of Decision trees; the difference lies in how the models are trained. While in Random Forest, all trees are built independently, XGBoost fits one tree at a time: first to all the outcomes in the training data and then to the residuals of the previous models. It computes second-order gradients of the loss function. L1 and L2 regularization parameters to penalizing model complexity are used. The number of trees, the trees' maximum depth, the number of features used in each split, and the L1 and L2 regularizations are the main parameters tuned for this model. XGBoost models were fit using the Sci-kit learn and XGBoost packages for python.

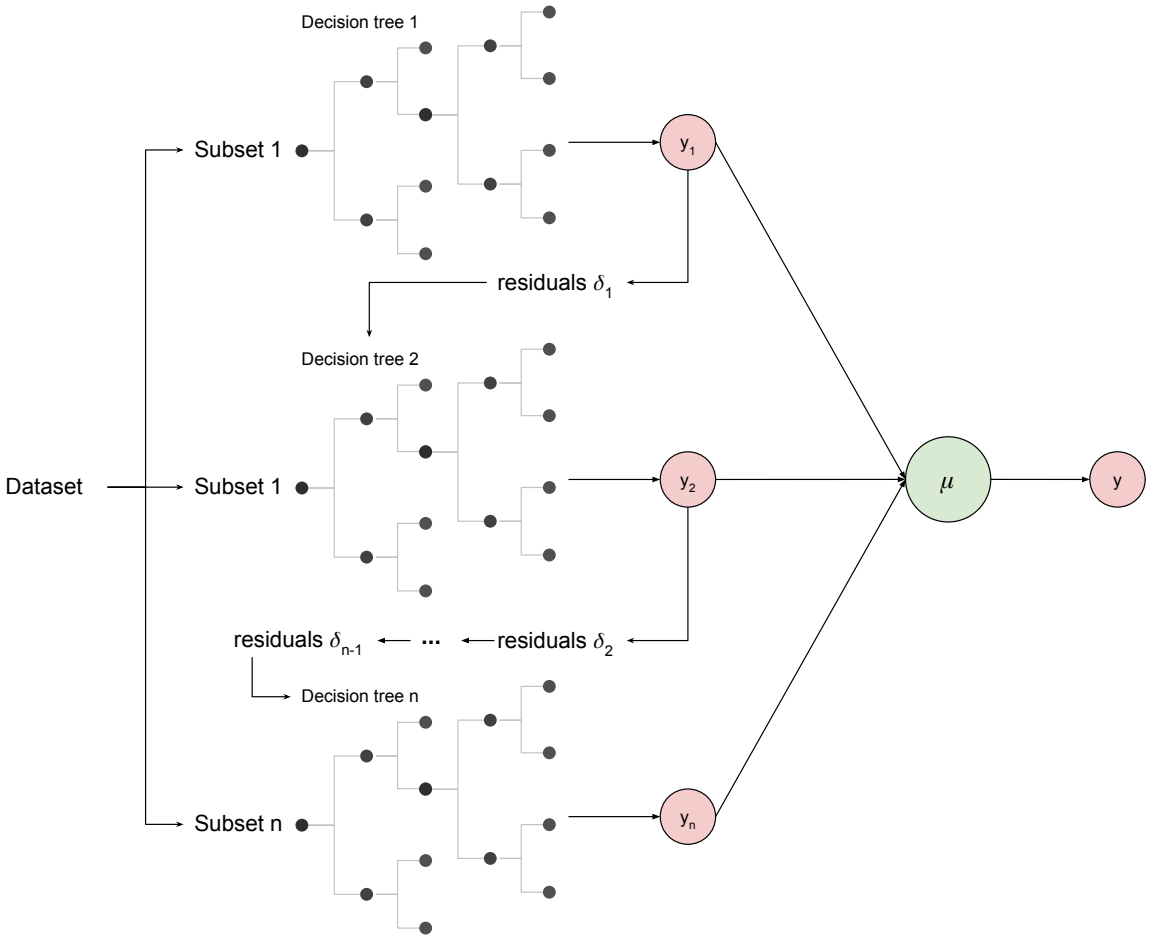


Figure 10. XGBoost Model Schema.

3.5. Model Evaluation

The models obtained are tested with two different set-ups. The first one is the common one in which all the samples created with the test set are used to evaluate the model using the model metrics explained below. In the results, this method is noted as standard set-up. The second set-up consists of a simulation of a real environment in which a silencing policy is implemented: once an alarm is triggered, alarms for the next three hours are suppressed. The goal of implementing a silencing policy is to avoid calling the rapid response team for consecutive alarms. In other words, the next three samples collected after an alarm is triggered are not used for testing. Furthermore, to not penalize the model when a true positive is silenced, if the silence period of the alarm triggered includes a true positive, the alarm triggered will be considered as a true positive too. In the results, this method is noted as silencing alarm policy set-up.

3.5.1. Model Metrics

The area under the Receiver Operating Characteristic curve (AuROC), the area under the Precision-Recall curve (AuPRC), and the specificity at 80% sensitivity (decision threshold) for the held-out test set are used as performance metrics to evaluate the models. To ensure that model is not biased by the randomized split, *5-fold Cross-Validation* (with a 90%-10% split within each fold) is used for training and testing purposes. It is reported the mean and the standard deviation. All the splits are performed following the stratification described in Section 3.2.5 Dataset Splits.

The MEWS and the NEWS are used as benchmarks. To compute the different metrics stated above, first, the value of the MEWS and NEWS for each sample is created. Then, each value is divided by the maximum possible score (14 and 20 respectively), so a value between 0 and 1 is created. Finally, using different trigger thresholds, the previous metrics are computed.

3.5.1.1 Metrics description

The Precision, also known as positive predictive value, is an evaluation metric that computes the rate of positives that are true positives. It can be calculated as follows:

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives} \tag{9}$$

In contrast, the rate of positives that are false positives is called false alarm rate (FAR) and can be computed as follows:

$$False\,alarm\,rate = 1 - Precision = \frac{False\,Positives}{False\,Positives + True\,Positives} \tag{10}$$

The Sensitivity or Recall, also known as true positive rate, describes how many events from the total have been labeled as positive. It is a useful metric when there is a high cost associated with a False Negative, which is the case of this work. The sensitivity decreases with the increase of precision. While the model should not miss impending events (high sensitivity), it cannot tolerate a low positive predictive value either. Thus, there is a trade of between the precision and sensitivity achieved by the model. It can be seen as the ability of the model to detect patients' events along the prediction window the model was trained for. It can be calculated as follows:

$$Sensitivity = \frac{True\,Positives}{True\,Positives + False\,Negatives} \tag{11}$$

The specificity, also known as true negative rate, describes how many negative samples from the total have been labeled as negatives. It gives a sense of how often a false alarm is triggered. It can

be seen as the ability of the model to reject healthy samples properly. It is computed as follows:

$$Specificity = \frac{True\,Negatives}{True\,Negatives + False\,Positives} \tag{12}$$

Analogously to the false alarm rate, the false positive rate (FPR) can be calculated by subtracting the true negative rate to the unit:

$$False\,positive\,rate = 1 - Specificity = \frac{False\,Positives}{False\,Positives + True\,Negatives} \tag{13}$$

The relation between the different metrics is schematized in the confusion matrix of Table 10.

		True Event			
		Yes	No		
Predicted Event	Yes	True Positive (TP)	False Positive (FP)	Precision TP / (FP + TP)	FAR FP / (TP + FP))
	No	False Negative (FN)	True Negative (TN)		
		Sensitivity TP / (TP + FN)	Specificity TN / (TN + FP)		
			FPR FP / (FP + TN)		

Table 10. Metrics relation.

3.5.1.2 Receiver Operating Characteristic curve

The Receiver Operating Characteristic curve is used to illustrate the sensitivity and the false positive rate for different decision thresholds. The false positive rate is plot in the x-axis, while the sensitivity in the y-axis. It represents a monotonously increasing function that goes from the (0,0) to the (1,1) points (from all samples classified as negatives to all samples classified as positives). Note that this statement is not necessarily true when the mean for multiple experiments is represented. The diagonal represents the random prediction, which means that the classification process is equivalent to classify all the samples randomly. The perfect classification is represented by a horizontal line from (0,1) to (1,1).

The area under the curve is the definite integral of the ROC curve. A perfect classifier will obtain an area under the curve of 1, a random classifier will be around 0.5, and a classifier which is not able to predict not even one true positive will have an AUC of 0. Thus, the higher the AUC, the better.

A graphic explanation can be found in Figure 11.

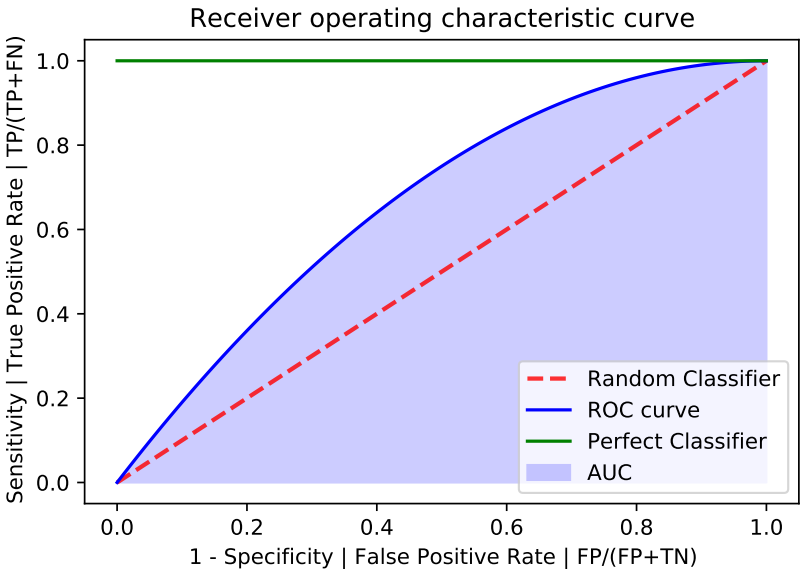


Figure 11. Receiver Operating Characteristic curve example.

3.5.1.3 Precision-Recall curve

Similar to the ROC curve, the Precision-Recall curve is a graphical plot of the precision against the recall for multiple decision thresholds. The Recall is plot in the x-axis, while the Precision in the y-axis. A perfect classifier will be represented by a horizontal line from (0,1) to (1,1) while a randomized one by a horizontal line in a lower height, that depends on the balance of positive and negative samples. It is a useful plot to use when the problem is defined by a highly imbalanced dataset.

The area under the curve is the definite integral of the PR curve. A perfect classifier will obtain an area under the curve of 1, and a random classifier will be equal to the label imbalance, as it will be represented by a horizontal line in the sensitivity level of the imbalance. Thus, the higher the AUC, the better.

A graphic explanation can be found in Figure 12.

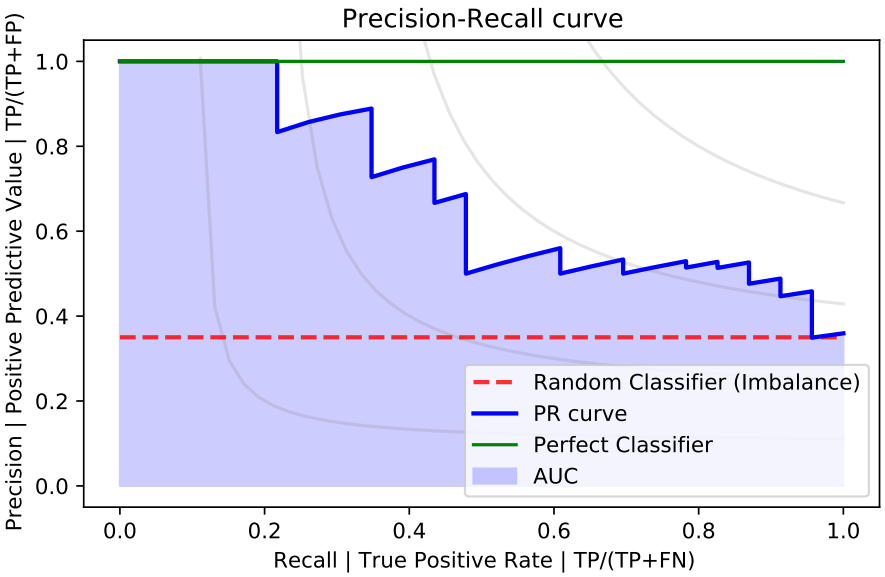


Figure 12. Precision-Recall curve example.

3.5.1.4. k-fold Cross-Validation

Cross-validation is a model validation technique useful to assess how the model will generalize in an unseen dataset. It can be also used to obtain the parameters that will perform better to an unseen set of data. In this technique, the model is trained in a subset of the dataset and then evaluated in a complementary subset which has not been used for training purposes. The goal of this method is to obtain an estimation of the performance of the model as well as to detect over-fitting.

In k-fold Cross-Validation, k different sets of training and validation or test set are created, always following the desired split technique, and making sure that the sets are not repeated. Then, for each set a new model is trained and evaluated. The results from the different sets are combined to obtain a more accurate and general performance metric of the model.

3.5.2. Hyperparameters Optimization

To prevent over-fitting, each model's parameters are initialized randomly and optimized on the training data using again a *5-fold Cross-Validation* algorithm with an 88.88%-11.11% split within each fold (80% training, 10% validation). All the splits are performed following the stratification described in Section 3.2.5 Dataset Splits.

The score of the validation set is the average of the score obtained by each fold. The combination of hyperparameters with the highest score are the ones used to create the model. As the model is evaluated also using a *5-fold Cross-Validation*, the hyperparameters are optimized on each evaluation

fold.

A Bayesian optimization using Gaussian processes algorithm is used to search along with the space of hyperparameters of each model. In this method, the machine learning model is approximate as a Gaussian probabilistic function and attempts to maximize (probability of improvement) its value (estimation + uncertainty) for a given space of parameters using the results obtained in the previous iterations. Then, the set of parameters that maximize the function are used to train the model and repeat the procedure.

3.5.3. Model Interpretability

Model interpretability is achieved by calculating different feature and signal importance scores for each type of model. For the logistic regression, just a mean decrease in accuracy (MDA) approach is used. For XGBoost and Random Forest, three different methods have been explored. First, the aforementioned MDA, second the Gini Importance [36] and finally the SHapley Additive exPlanations (SHAP values) [37].

In MDA (Permutation importance), the model is tested permuting the values of one of the features at a time. The procedure is repeated for all the features that compound the original model. This method is based on the hypothesis that if a model depends on the feature shuffled, it will affect negatively the model performance, as the association between the feature and the outcome is removed. The more that the model performance metric (in this work, AuROC) drops when a feature is shuffled, the more important it is. So, the score for each feature is the difference between the AuROC when the model is tested without permuting any feature and the AuROC when the model is tested permuting that one. This method is biased towards independent features; as for correlated features, the model performance will not drop much when one of the features is removed. To assess signal importance using this method, the same procedure is applied but permuting all the features related to one signal at the same time. To remove the influence of the randomized shuffle, ten iterations have been done per feature and signal, as result is reported the mean of these ten iterations.

Gini Importance explanation calculates the feature importance using the impurity decrease of the nodes. The variable importance is the sum over the impurity decrease of all splits conducted on the feature (across all trees), weighted by the probability of reaching that node and normalized by the number of trees. To assess signal importance with this method, the sum of the score value of the features related to each signal is computed. Remember that the Gini impurity at a given node t can be computed as:

$$GI(t) = \sum_{j=0}^{J} P_{jt} \cdot (1 - P_{jt}) \tag{14}$$

where j are the different classes, J the number of unique classes, and P_{jt} the class j frequency at node t. The impurity decrease can be calculated as the difference between the node's impurity and the weighted sum of the impurity's child nodes.

SHAP values explanation defines how much the specific value of each feature drove a specific prediction. In other words, it defines feature importance for each prediction. Each feature is assigned an impact to the model prediction that can be positive or negative (accordingly if it makes the prediction increase or decrease). With this method, it is possible to assess not only the feature importance for the model overall, but also for each prediction the algorithm is making, which might give important insights to the medical team when a real time implementation is performed.

4. Results

4.1. Overall Prediction Performance of Best Models

The combination of parameters that result in the best performance is using a four hours prediction window, no gap, four hours vitals look-back window, and missing indicators for both, vitals, and laboratory values. The ROC and PR curves for the models developed with this layout are shown in Figures 13 and 14 for the standard and the silencing alarm evaluation set-ups respectively.

As there is interest in extending the prediction horizon as much as possible, the ROC and PR curves for eight hours prediction window are also shown in Figures 15 and 16.

In Table 11 is summarized the results obtained for the top 5 models according to the AuROC value obtained for each type of evaluation method without considering model type.

The following sections are discussed just on the standard set-up (except for the Impact of Gap Window Timing on Performance), as the take-aways from the silencing alarm policy are exactly the same. The results for the standard set-up are found and discussed in each section while the results for the silencing alarm policy can be found in Appendix A: Supplementary Material.

Model Type	Gap	Prediction Horizon	Vitals look-back window	Missingness encoding	Set-up	AuROC	Specificity (80% Sensitivity)
XGBoost	0	4	4	Both	Silencing policy	0,832	0,707
XGBoost	0	4	8	Both	Silencing policy	0,829	0,707
XGBoost	0	8	4	Both	Silencing policy	0,798	0,656
XGBoost	0	8	8	Vitals	Silencing policy	0,789	0,646
XGBoost	0	8	8	Both	Silencing policy	0,787	0,646
XGBoost	0	4	8	Both	Standard	0,747	0,535
XGBoost	0	4	4	Both	Standard	0,746	0,535
XGBoost	0	8	4	Both	Standard	0,738	0,515
XGBoost	0	8	8	Vitals	Standard	0.724	0.515
XGBoost	0	8	8	Both	Standard	0.723	0.505

Table 11. List of top 10 models according to AuROC value.

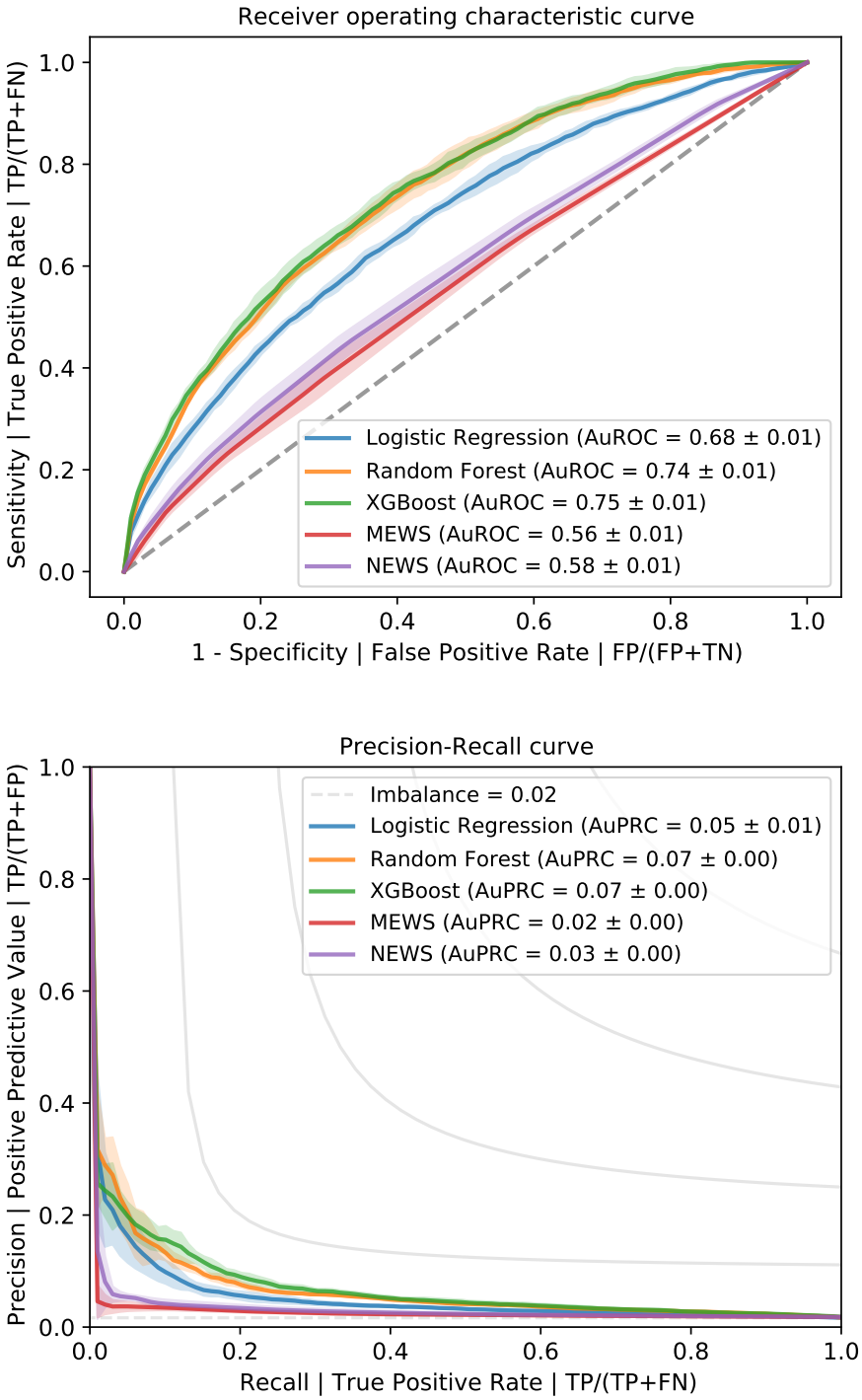


Figure 13. Model performance. ROC and PR curves. Standard set-up. 4 hours prediction.

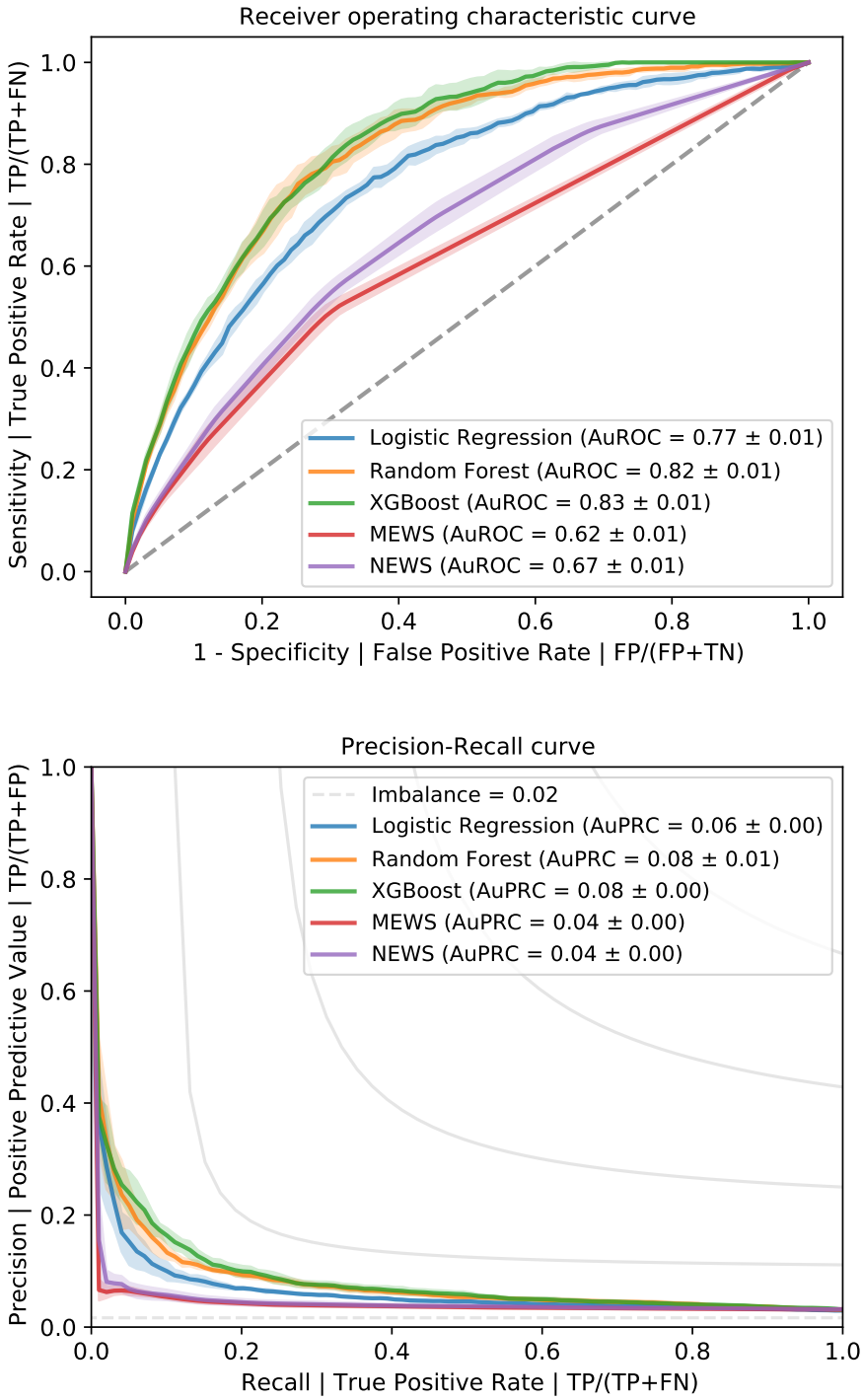


Figure 14. Model performance. ROC and PR curves. Silencing alarm policy. 4 hours prediction.

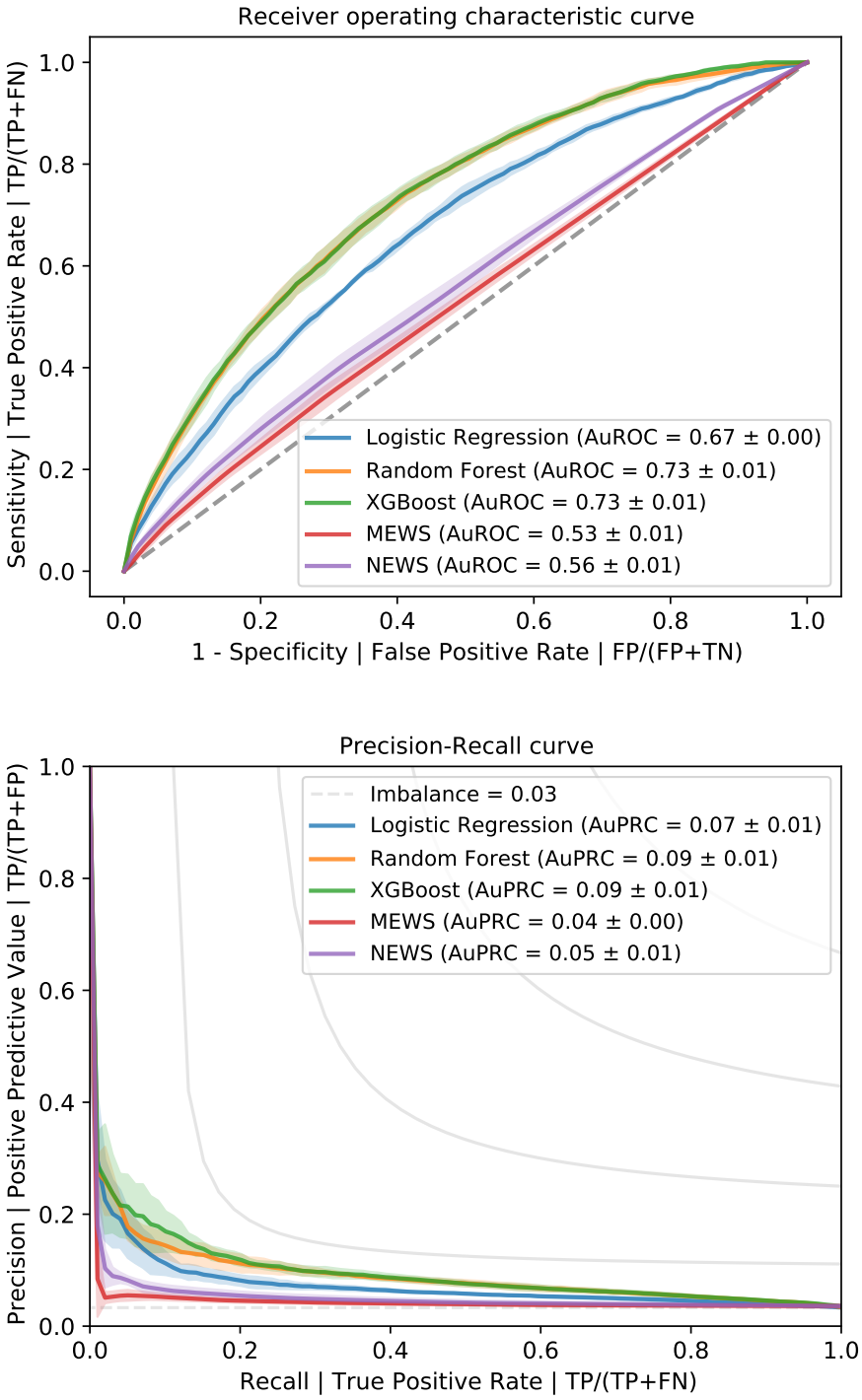


Figure 15. Model performance. ROC and PR curves. Standard set-up. 8 hours prediction.

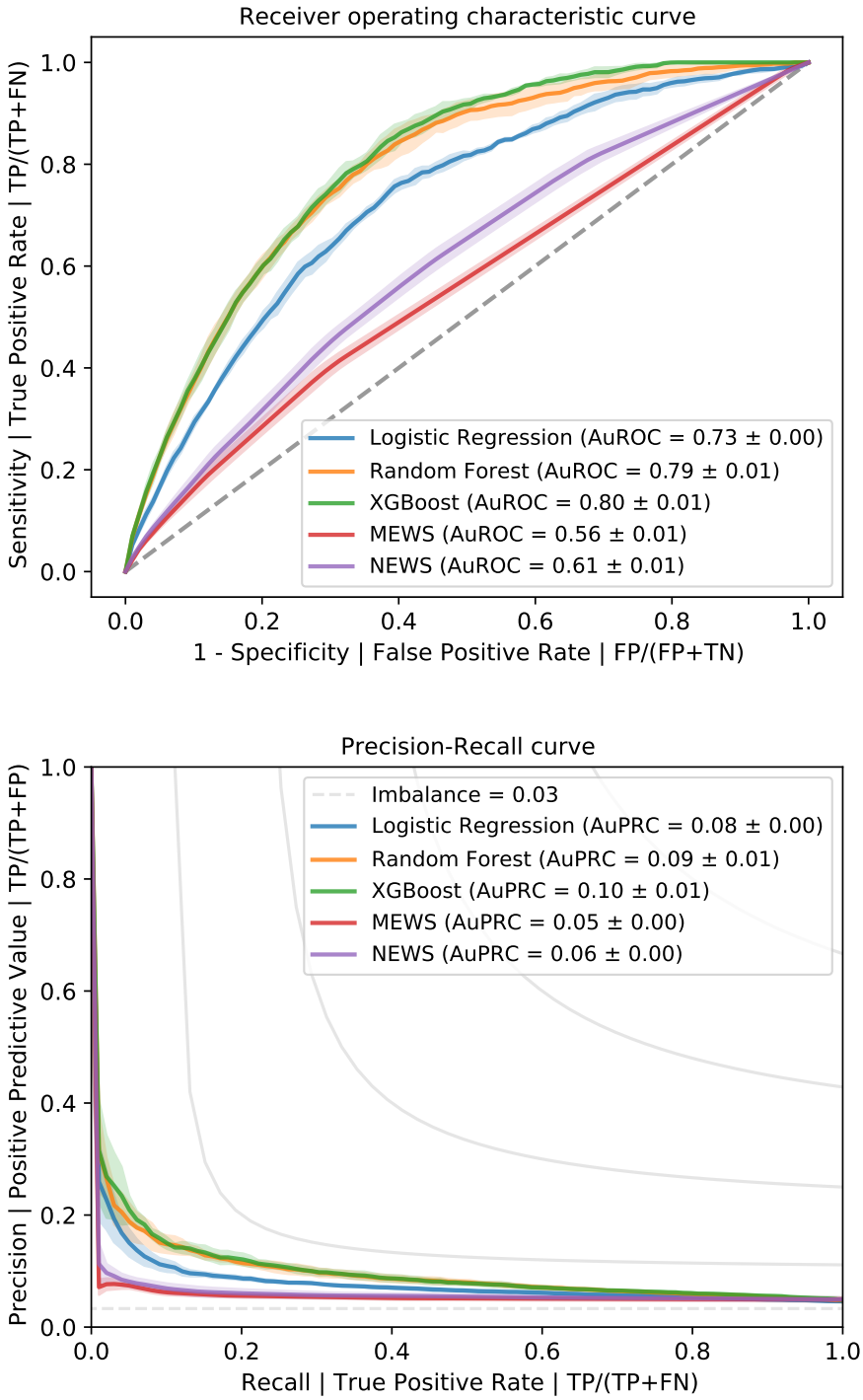


Figure 16. Model performance. ROC and PR curves. Silencing alarm policy. 8 hours prediction.

4.2. Impact of Data Window Timing on Performance

The impact of the length of the vitals look-back window is studied with models using eight hours prediction window with no gap and using all the features. Figure 17 shows the results for the standard set-up. Note that while for tree-based models the performance decreases slightly with the length of the vitals look-back window (there is no statistical difference), for the logistic regression the performance increase with it.

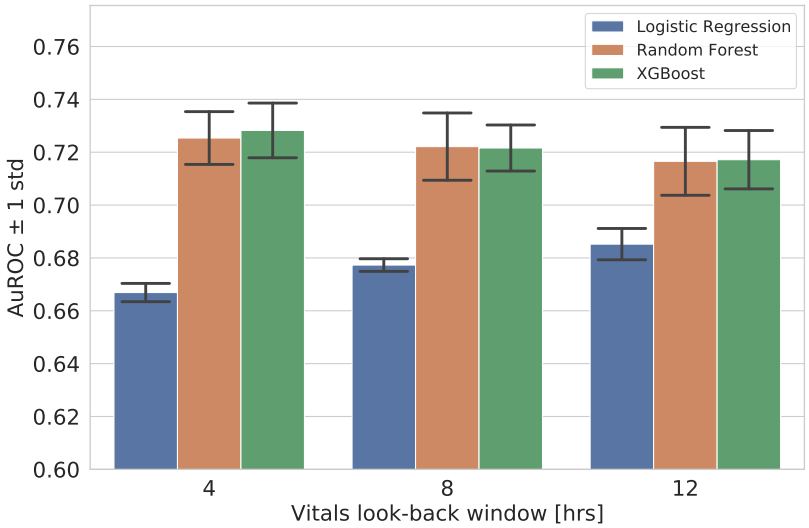


Figure 17. Model performance varying vital signs look-back window. Standard set-up.

4.3. Impact of Prediction Horizon on Performance

Using no gap window, eight hours look-back window for vitals, and all the features, the impact of different prediction horizons on performance is assessed. Figure 18 shows how performance decrease with the increase of the prediction horizon. When a twenty-four hours prediction window is used, the MEWS performs as a random classifier.

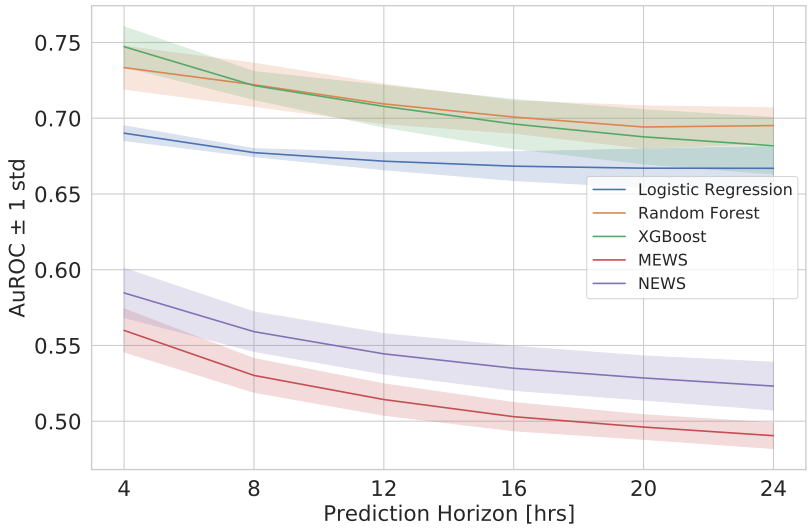


Figure 18. Model performance varying prediction horizons. Standard set-up.

4.4. Impact of Gap Window Timing on Performance

The performance when a gap window is used is compared with models using an eight hours prediction window, eight hours look-back window for vitals, and all the features. As expected, the use of a gap window affects negatively the performance of the model with the standard evaluation set-up. However, this is not the case for the silencing alarm policy, as there is no change in the performance when increasing the gap window.

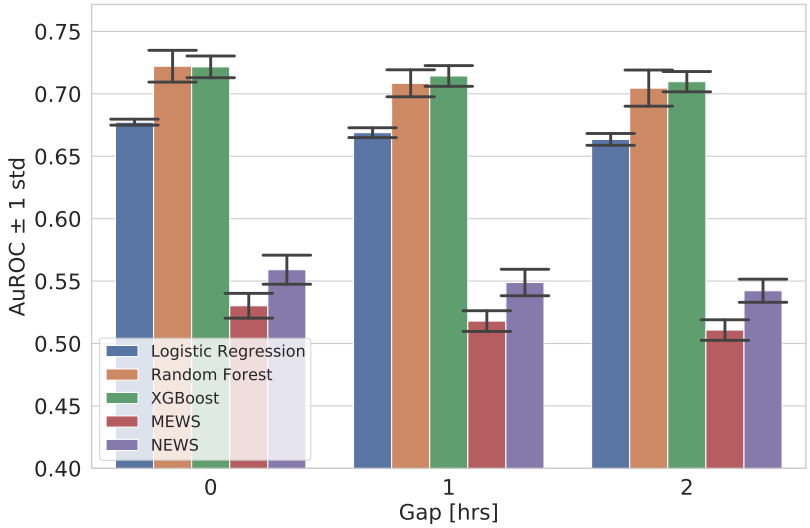


Figure 19. Model performance varying gap window. Standard set-up.

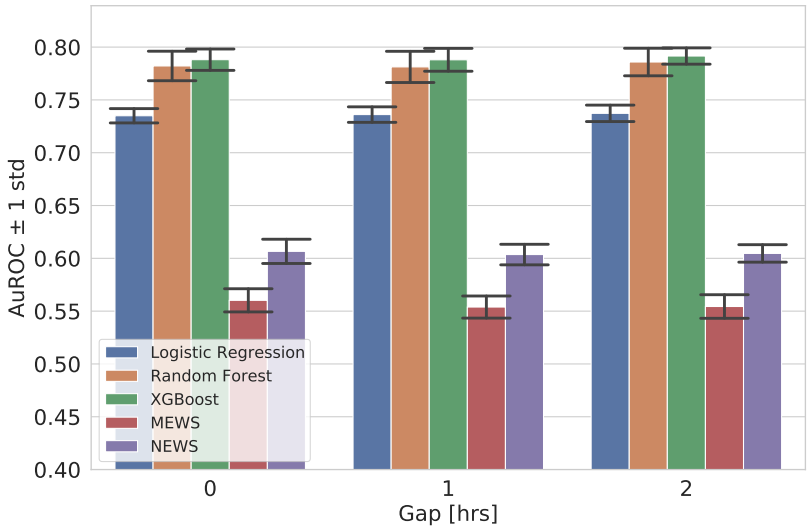


Figure 20. Model performance varying gap window. Silencing alarm policy.

4.5. Impact of Missingness and Sampling Frequency Encoding of Data from the Window

The performance of the inclusion/exclusion of the vital feature *count* and the missing flag for the laboratory values is performed across models with an eight hours prediction window, no gap, and eight hours look-back window for vitals. Figure 21 shows AuROC ± 1 standard deviation when both features are used, just one of them or none. It is noticeable that the *count* feature contains important predictive information while the missing flag not.

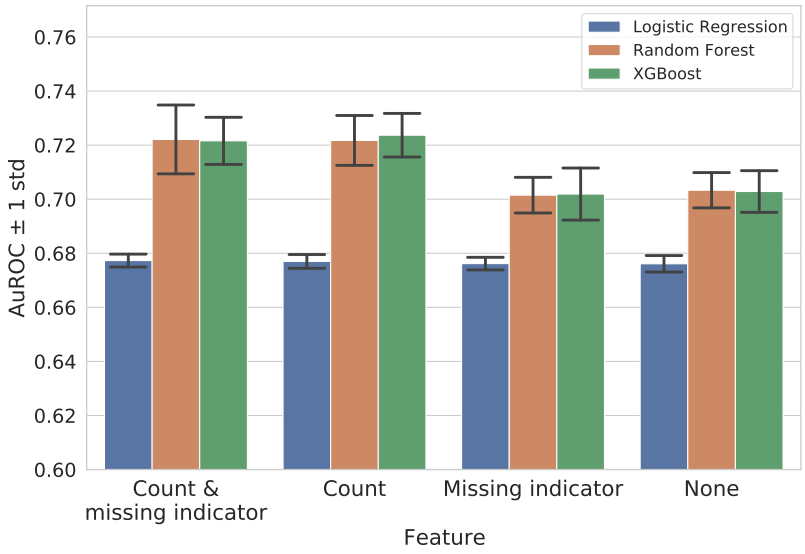


Figure 21. Model performance with and without missingness and frequency indicators. Standard set-up.

4.6. Impact of the Algorithm Choice

The performance of the different algorithms is compared using an eight hours prediction window, no gap, eight hours look-back window for vitals, and all the features. Figure 22 shows the AuROC ± 1 standard deviation for the different models. The models computed in this work outperform the early warning systems that are used as benchmarks. The tree-based models perform better than the logistic regression. There is no statistical difference between XGBoost and Random Forest.

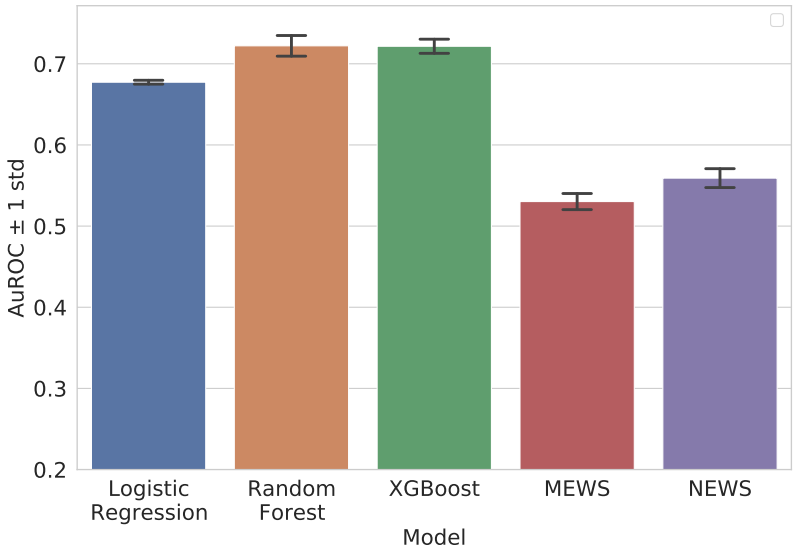


Figure 22. Model performance varying algorithm. Standard set-up.

4.7. Variable Importance

To analyze the feature and signal importance of the models, the set-up where an eight hours prediction window with no gap has been used. In this section, just feature and signal importance for an XGBoost model is analyzed. The results for Random Forest and Logistic Regression are attached (but not discussed) in Figures A.5 to A.10 from Appendix A: Supplementary Material.

The feature importance (Figures 23 and 25) complements the results from the Impact of Missingness and Sampling Frequency Encoding of Data from the Window analysis, as using both methods, the count feature for different signals is among the 15 most important features. Furthermore, the missingness indicators of the laboratory values are not throughout the most important features for any of the models/methods.

Despite the methods are quite different, thirteen out of the fifteen most important signals (Figures 24 and 26) are the same for both methods (different order). Similar for the features (Figures 23 and 25), eleven out of fifteen are shared between methods.

Note that for both methods all the vital signs are included as important signals for the model except for the Glasgow Coma Scale in the Gini criterion.

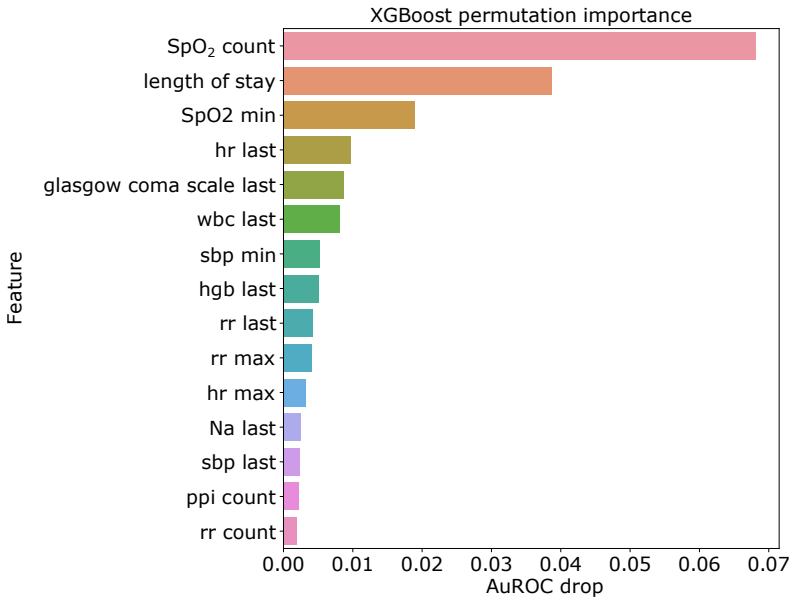


Figure 23. Feature importance based on permutation importance using XGBoost.

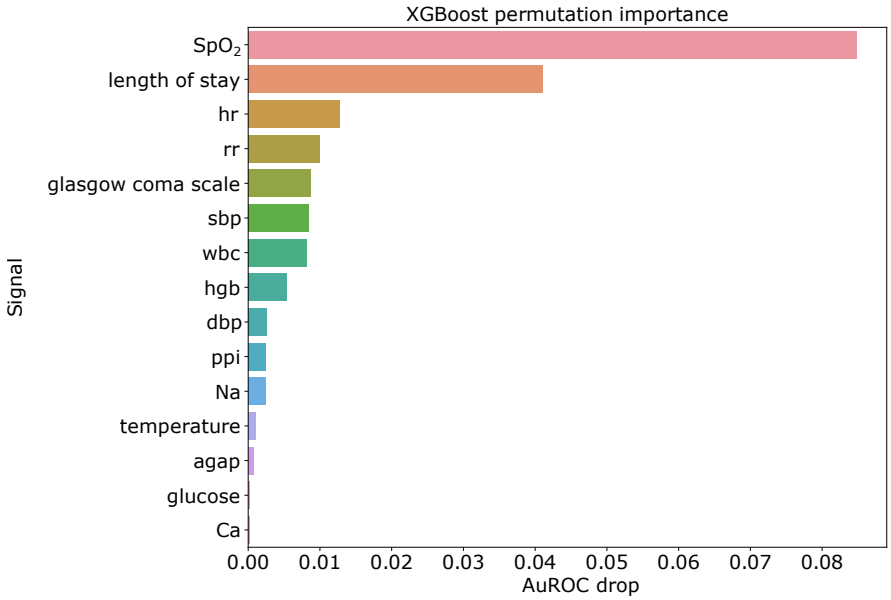


Figure 24. Signal importance based on permutation importance using XGBoost.

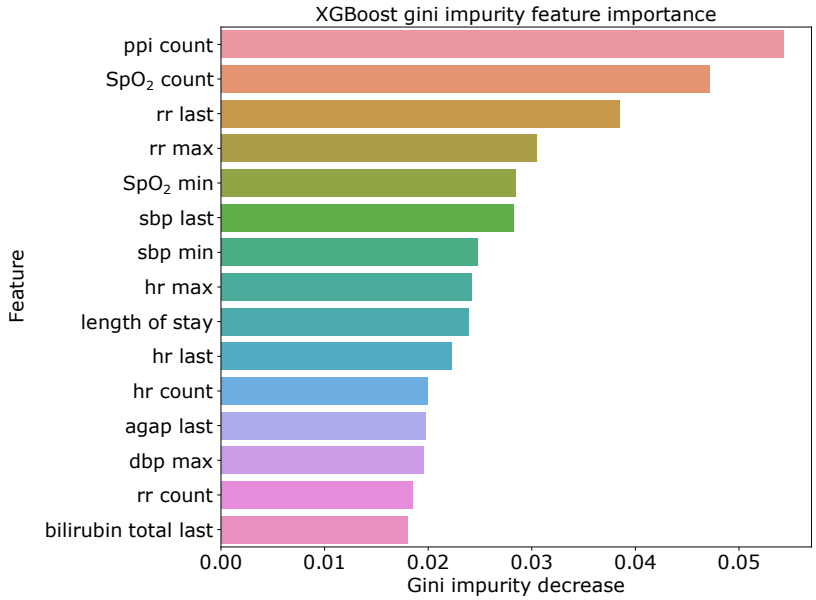


Figure 25. Feature importance based on Gini criterion using XGBoost.

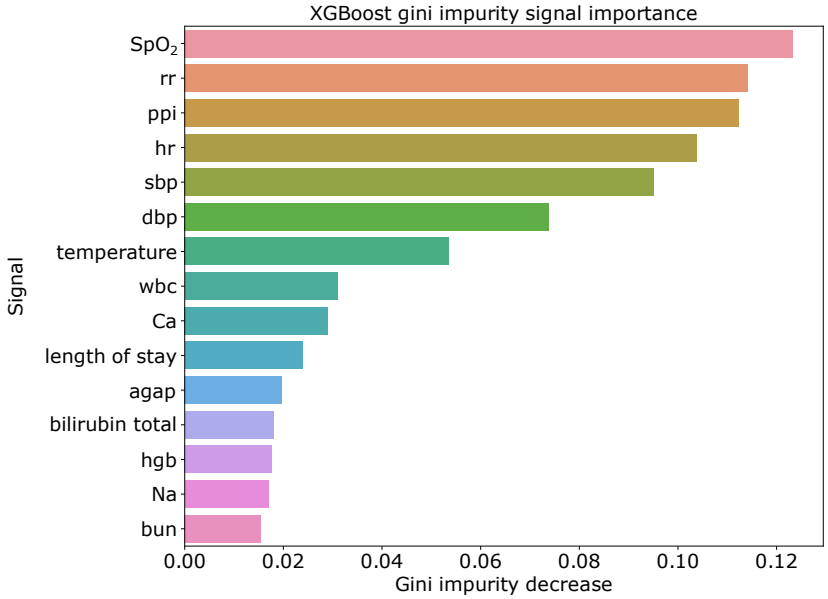


Figure 26. Signal importance based on Gini criterion using XGBoost.

5. Discussion

5.1. Interpretation of Experimental Results

The model chosen to implement the final early warning system is the XGBoost using an eight hours prediction window, no gap, four hours of look-back window for vitals, and all the features. Note that the precision at 80% Sensitivity (decision threshold) is 7% with an AuROC of 0.798 when the silencing alarm policy is used. This means that the false alarm rate is about 93%. According to the literature, it is a common rate in early warning systems but high enough to lead caregivers to miss the true positive alarms and provoke alarm fatigue [38]. Note that despite it might be an acceptable false alarm rate, it is clearly one of the drawbacks of the model presented that needs to be improved.

In Figures 27 and 28, is shown examples of clinical vitals trajectory together with the model prediction and the top contributing features using SHAP values. *Feature high* plot lists the five features with the highest SHAP value, that is, that pushes the prediction to a higher value; *feature low* lists the five features with the lowest SHAP value, that is, that pushes the prediction to a lower value. The evolution is presented from forty-eight hours after admission until the time of the event. For clarity, just five relevant features of each type are shown every ten to eight-teen hours. Four more examples of patient trajectories are shown in Appendix A: Supplementary Material, Figures A.11 to A.14. Each one of the Figures corresponds to a different patient from the test set.

These Figures can be seen as illustrative examples of continuous monitoring of a patient using the algorithm implemented (real-time implementation). Note that for all the examples shown, the algorithm detects the adverse event several hours in advance. However, in Figure 27 the decision threshold is surpassed a long time before the eight hours prediction horizon, while in Figure 28 several false alarms are triggered every day. It is an expected behavior given the high % of alarm rate.

For these reasons, it is believed that the performance of the models can be improved by using a more complex implementation policy, for instance using a cumulative risk prediction instead of just the risk predictor itself. Also, the use of a different evaluation method where a constant increase of the prediction risk that triggers alarms before the prediction horizon window (Figure 27) are not penalized should be considered [39].

Furthermore, a similar deployment plan to the one described in [20], where a remote nurse team decides whether to act upon the alarm or not by continuously monitoring remotely the records of patients who have been identified at risk of suffering a code blue or rapid response, will help to reduce the negative impact of such high false alarm rate.

The performance obtained with the silencing alarm policy is considerably higher than the one obtained using the standard evaluation. This phenomenon is attributed to the reduction of the false positive rate. As it can be seen in the patient trajectory from Figure 28, during 78 to 112 hours after admission, the algorithm predicts wrongly the occurrence of an event for 34 consecutive hours. With the silencing alarm policy, the number of false positives is reduced by just triggering one-fourth of the alarms. Similar behavior can be appreciated in each of the sample trajectories shown in the work. Thus, in a clinical set-up, is highly recommended to use a similar policy to the silencing alarm one, to avoid trigger too many consecutive alarms.

The fact that the models created are not affected by the gap window confirms that there is no label leakage as the models are not biased towards samples close to the event (Figures 19 and 20). Furthermore, the invariability on performance when a silencing alarm policy is used is telling that 1) The decrease in performance when the standard evaluation is performed is mainly due to the removal of true positive samples (making greater the imbalance), and not because the model trained performs worse; 2) Most of the events that are detected are predicted with at least two hours in advance (maximum gap tested). This last statement is corroborated with the multiple examples of patient trajectories shown in this report: Figures 27, 28, A.11, A.12, A.13, and A.14.

As hypothesized, basic machine learning models can outperform the common early warning systems such as MEWS and NEWS (Figure 22). Furthermore, tree-based models also show a considerably better performance than Logistic Regression. Despite most researches tend to use Logistic Regression models instead of tree-based models for their interpretability, in this work this issue is overcome by performing an exhaustive analysis in feature and signal importance using different methods.

The implementation of SHAP value helps to achieve model interpretability. When used in a real-time implementation, they give some insights about what is irregular in the patient physiology when an alarm is triggered. Altogether with the importance of the different features and signals shown in Section 4.7. Variable Importance allows the user to interpret and understand the predictions of the model. Furthermore, the methods implemented can be also used in black-box model types such as deep neural networks.

Unfortunately, with the results obtained in the impact of look-back window for vital signs is not possible to draw any conclusions (Figure 17). It is suggested to use a wider variety of length windows as well as with different combinations of set-ups to be able to obtain demonstrative results.

Finally, in Figure 21 it is shown the predictive power of the *count* feature. In contrast, the missing indicator for laboratory values seems to be useless, however, this result might be biased by the length of the look-back window, so it is suggested to test the impact of the inclusion/exclusion of this feature

with different lengths of look-back window for laboratory values.

5.2. Limitations

As the model is trained and tested with data corresponding to patients who sooner or later had an arrest, the behavior of the algorithm developed is unexpected in front of patients who are not at risk of code blue or rapid response events. Thus, the model has to be tested and/or retrained with a set of patients who never had an event, as during clinical implementation it will be used for patients that might not have one of these events during their hospitalization.

In the same line, no external validation is performed, so there is no guarantee that the algorithm implemented can work for an institution different than MGH. Therefore, the utility of this model is limited to MGH, but the methods used to implement it might be useful for other hospitals to develop their own model. Furthermore, the count feature might influence the model in a completely different way in another institution.

Despite it is shown that the machine learning models implemented outperform some of the most common early warning system to predict code blue and rapid response events, it is important to bear in mind that these warning systems are created to detect any kind of clinical deterioration, so some of the samples labeled as False positive might be detecting other kinds of outcomes that are not included in this work, and consequently the performance shown in this work is lower than the real one. Further steps include adding some of these outcomes such as transfer to the ICU to see if 1) it is possible to improve the performance of the current model, 2) compare the performance with the early warning scores in a more equitable set-up.

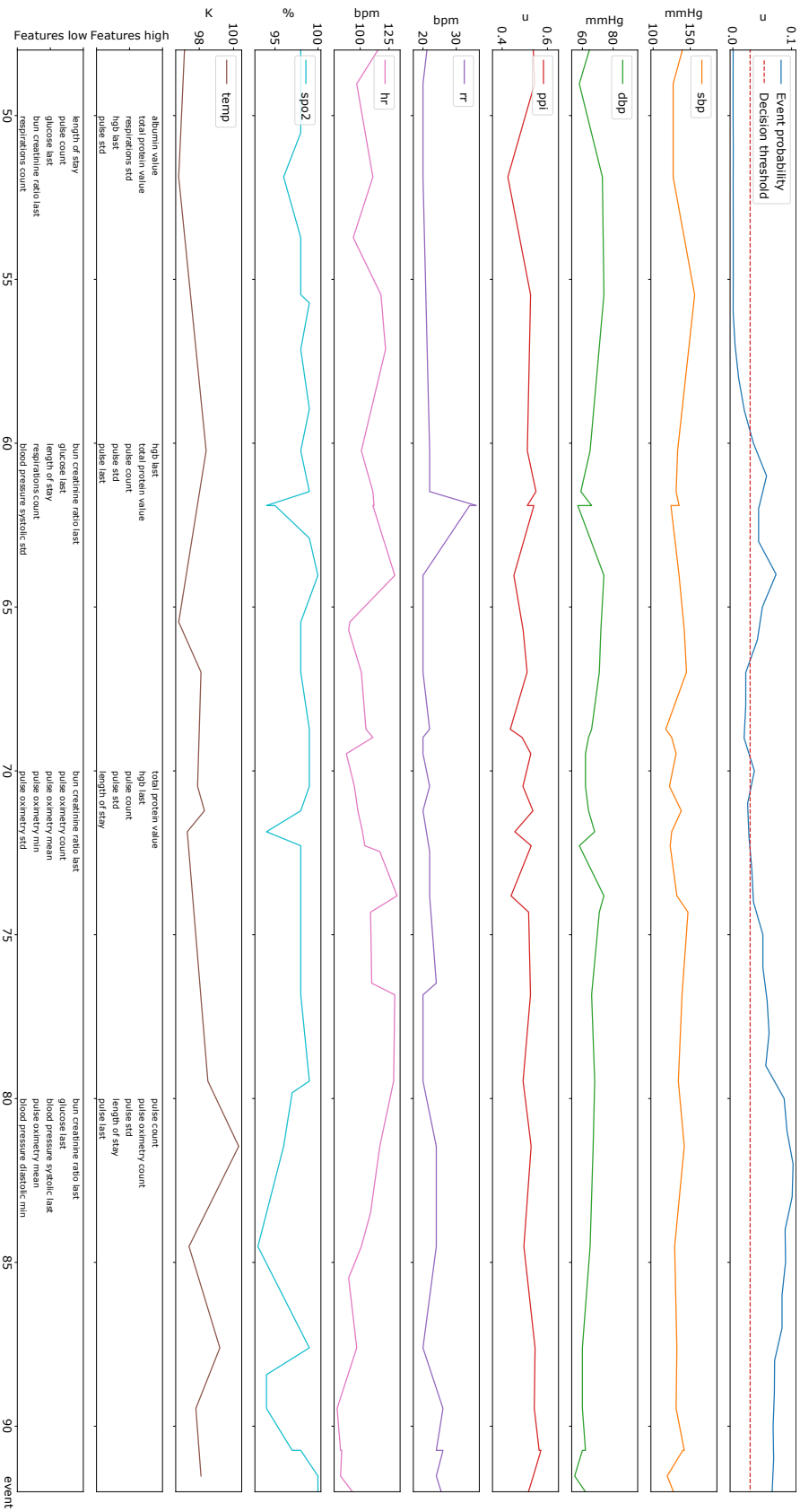


Figure 27. Illustrative example of a patient's trajectory 1. Rapid response Event.

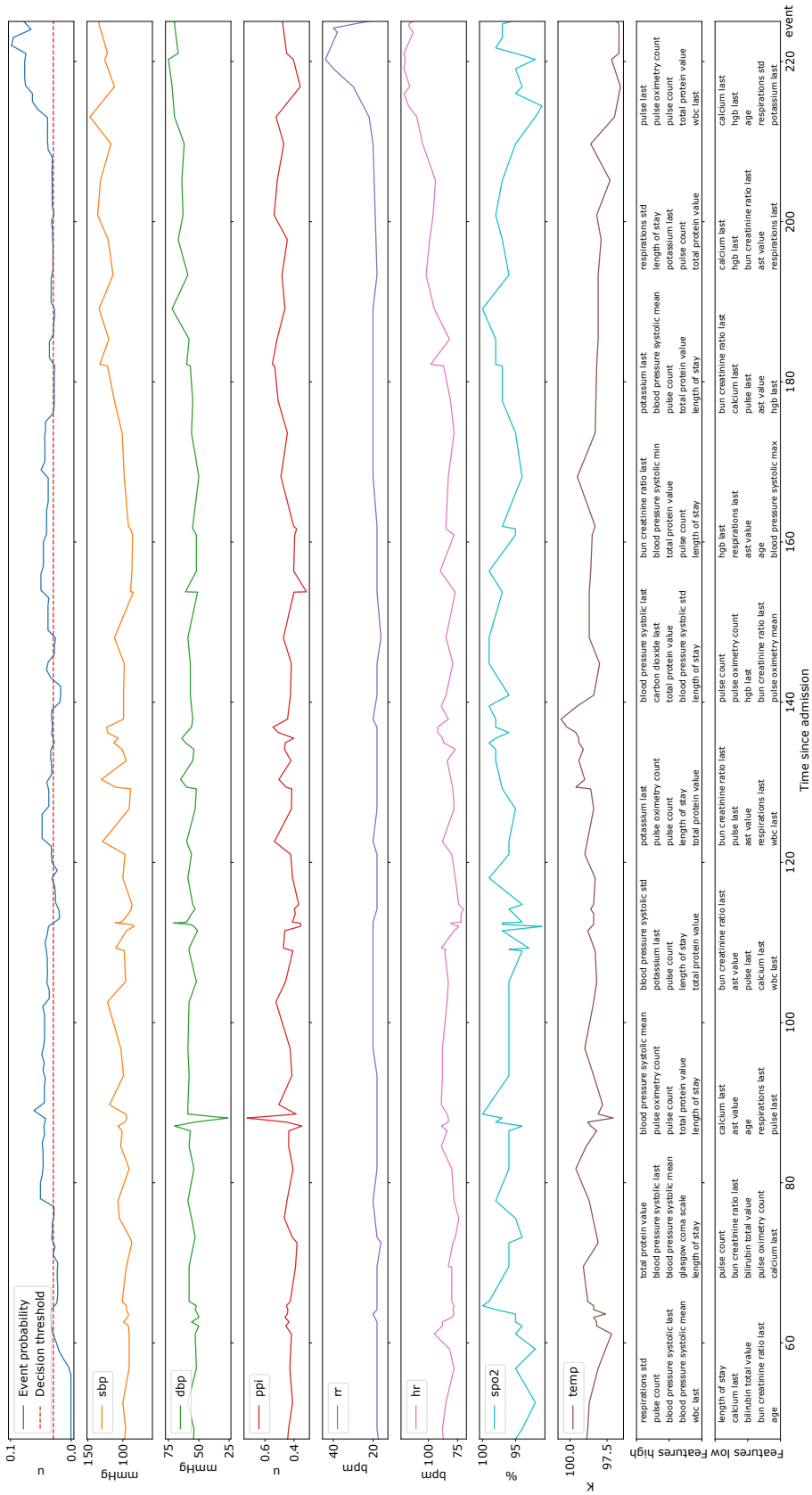


Figure 28. Illustrative example of a patient's trajectory 2. Code blue event.

6. Project Timeline

The development of the project has been divided into 6 different phases: *Conception and Initiation*, *Literature Review*, *Data Acquisition*, *Data Pre-processing*, *Algorithm Implementation*, *Results Dissemination*, and *Thesis Manuscript and Defense*. Each project phase is compound by several sub-phases. The project has had a duration of third-teen months, in the timeline between the 1st of April of 2020 to the 30th of April of 2021. The timeline has been divided into sets of half month (two weeks sprint), and it has been assigned at least one project phase to each set. The duration of each phase and the months that were tackled is depicted in the Gantt chart in Figure 29. Note that an additional week after the submission of this report is added in order to prepare the defense of the thesis.

Most relevant information regarding *Conception and Initiation* and *Literature Review* phases are described in Sections 1. Introduction and 2. Objectives. *Data Acquisition* phase is described in Section 3.1. Clinical Data Collection and Organization; all the Aguirre-Lab members made important contributions in this phase. Section 3.3. Data Pre-processing contains all information related to *Data Pre-processing* phase. *Algorithm Implementation* phase content can be found in Sections 3.2. Prediction Problem, 3.4. Prediction Algorithms, and 3.5. Model Evaluation. Finally, the *Results Dissemination* phase is described in Sections 4. Results and 5. Discussion.

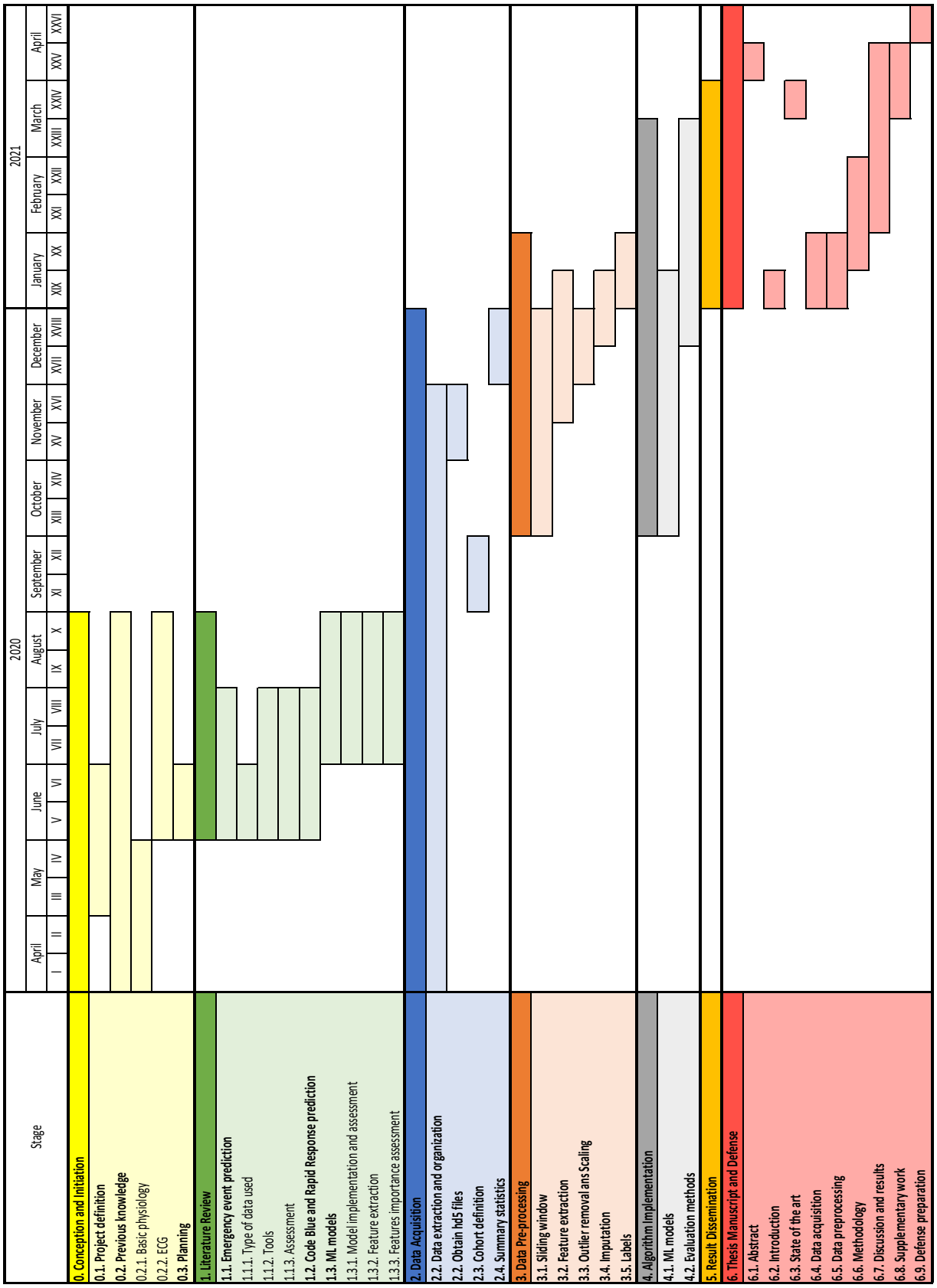


Figure 29. Gantt chart of the project timeline.

7. Economic, Social, and Environmental Impacts

The project's environmental impact during its development phase is negligible since it was carried out based on simulation. Despite there is no state nor federal legislation for electronic waste in the state of Massachusetts neither in the United States, to minimize the impact, when the life cycle of the different hardware elements is over, these will be disposed to the IS Service Desk of the MGH according to the policies and practices by the Mass General Brigham association (ISPR-8a.4: Enterprise Secure Media Destruction Procedures).

It is assumed that until the research is driven to develop a medical product, it is not possible to quantify the economic impact. Nevertheless, in the next Section, 8. Budget the project cost is estimated.

As stated in the introduction, cardiac and respiratory arrests are the main diseases related to code blue and rapid response alarms. Not only have associated a high mortality rate, but also people who survive develop significant disability and medical conditions such as neurological and cognitive sequels [40]. This work helps to make a step forward to mitigate such consequences by improving the early detection of the events, which allows the medical team to prevent the event or at least, reduce the adverse effects.

Besides, it is hypothesized that machine learning-based early warning systems (similar to the one developed in this work) will have a social and economic impact worldwide once deployed. On the one hand, it will help to provide affordable health care to everyone by reducing costs. Moreover, such technologies will help to provide medical support, especially to those communities with reduced medical staff/known-ledge. In a clinical set-up, not only helps to prioritize patients, but also to focus on the abnormal features of the patient physiology [41]. Furthermore, such systems will help to create a less overwhelming environment for the clinicians by, for example, reducing the false alarm rate.

It is important to mention that the approval of this work by the IRB ensures minimal harm and ethical practices for the use of identifiable data.

8. Budget

In this chapter, the cost of the human, hardware, and software, and network resources are analyzed. Each type of resource is described in a different subsection. The budget is calculated as if the project was run by a research group without students involved (all members are salaried, and no student licenses are considered).

8.1. Wages

The research group comprises four engineers, one post-doctoral researcher, and two MD, Ph.D. leading the group (Principal Investigators). One engineer was working full-time on this project, the rest of the team was spending just 10% of their time on the project.

Discriminating by role and taking into account the project duration of 13 months (271 working days), the total costs associated with the workers' wage are presented in Table 12.

Role	Wage (per month) [$]	Work load [%]	Total wage
Engineer 1	2.834,00$	100	36.842,00$
Engineer 2	2.834,00$	10	3.684,20$
Engineer 3	2.834,00$	10	3.684,20$
Engineer 4	2.834,00$	10	3.684,20$
Post-Doctoral Researcher 1	4.251,00$	10	5.526,30$
Principal Investigator 1	12.500,00$	10	16.250,00$
Principal Investigator 2	12.500,00$	10	16.250,00$
Total	-	-	85.920,90$

Table 12. Human resources costs.

8.2. Hardware and Software

As it is a computational project, a unique customized workstation plus additional hardware to enable work from home has been used. All components are brand new, so no depreciation has been applied yet. The estimated depreciation for each computer component used during the project is shown in Table 13. The depreciation of a component is computed according to the formula in Equation 15, based on a linear depreciation.

$$\text{Depreciation} = \frac{\text{Units} \cdot \text{Unit Price} \cdot \text{Hours of Use}}{\text{Depreciation Period} \cdot 250 \text{ working days per year} \cdot 8 \text{ hours a day}} \tag{15}$$

The depreciation period is equal for all the equipment, and is set to 5 years, following the recovery periods set by the federal agency from the United States of America, Internal Revenue Service (IRS)

[42]. All the components have been used the same amount of time, and it is estimated to be around 8 hours per working day.

Resource	Units	Unit Price	Depreciation Period	Hours of Use	Depreciation
Monitor	2	231,00$	5 years	2168	100,16$
Keyboard and mouse	1	18,23$	5 years	2168	3,95$
Case	1	94,99$	5 years	2168	20,59$
Motherboard	1	159,99$	5 years	2168	34,69$
SSD	1	319,99$	5 years	2168	69,37$
HDD	1	194,99$	5 years	2168	42,27$
RAM	1	169,99$	5 years	2168	36,85$
GPU	1	1.625,00$	5 years	2168	352,3$
Processor	1	278,99$	5 years	2168	60,48$
Total	-	3.324,17$	-	-	720,68$

Table 13. Hardware resources costs.

Regarding the software resources, most of them (Ubuntu, Python, SQL, and Latex) are open-source software thus, free. The unique one to consider is MATLAB®, which a unique Academic, Individual Use, Annual license has been used, which has been totally depreciated, as the license has expired. A GitHub pro account is used, which can be created with no additional cost using the academic email from MGH, thus, the cost of the account is considered part of the network resources. The total cost associated with the software resources is shown in Table 14.

Resource	License Price
Ubuntu	0$
Python	0$
SQL	0$
Latex	0$
MATLAB®	250$
Total	250$

Table 14. Software resources costs.

8.3. Network Resources

Facility costs such as building depreciation, maintenance, utilities, etc. and administrative costs such as hospital research support programs and administration, management, etc. incurred at MGH are charged according to the rates specified in the Facility and Administration Rate Agreement from MGH that can be found in Appendix B: F&A MGH Rates.

As this research has been carried out in an On-Site facility from MGH, it corresponds to a rate of 68% of the direct costs with the exception of the hardware. As it is shown in Table 15, the total cost

of network resources is 58.596,21$.

8.4. Summary

In conclusion, the total cost of human, hardware, software, and network resources, is summarized in Table 15. The total cost of the project is 118.740,30$.

Resource	Associated Cost
Direct Cost	
Wages	85.920,90$
Software	250$
Hardware Depreciation	720,68$
Total without Hardware Depreciation	86.170,90$
Total Direct Cost	86.891,58$
Indirect Cost	
Network	58.596,21$
Total Indirect Cost	58.596,21$
Total Cost	145.487,80$

Table 15. Overall project costs.

9. Conclusions and Future Work

In the course of this thesis, a functional early warning system for code blue and rapid response prediction have been developed. After an analysis of the clinical problem and the state of the art the aim of the thesis is defined. The goal of the model developed is to detect patient deterioration that leads to a code blue or a rapid response in a clinical setting. Such prediction is presented in the form of a risk score that triggers an alarm when a decision threshold is surpassed. This predictor aims to help clinicians determine which patients are at risk to prioritize their work and prevent code blue and rapid response events. For this reason, achieve model interpretability is a requirement. To do so, an exhaustive analysis of the feature and signal importance is performed. The impact of the different parameters to define the training and test features as well as two evaluation methods are also explored.

The analysis performed in this work yield the following results related to the specific aims stated in Section 2. Objectives, Scope, and Departure Point:

- It has been built a database composed of all the patients who had a code blue or a rapid response at MGH. The modularity of the pipeline developed can be used to coalesce data for new cohorts of patients for different research purposes. Moreover, it collects the necessary data to step on the next phases of the project.
- It is demonstrated the power of the physiological data to predict cardiac arrest. More specifically, it is shown the importance of vital signs as predictors during the twenty-four hours period prior to the emergency event. Furthermore, it is shown that the sampling frequency of the vitals is a strong predictor for such events.
- A functional early warning system based on XGBoost is proposed in this work. This model outperforms traditional methods like logistic regression and the most common early warning systems MEWS and NEWS. However, the high false alarm rate triggered by the system compromises the implementation in a clinical setting. Furthermore, it is demonstrated that using an evaluation approach closer to the one practiced in a clinical setting, helps to obtain better performance metrics.

From this point, the roadmap of the project is divided into different investigation lines, all of them with the aim to improve the performance of the proposed early warning system and answer the open questions from the state of the art. More specifically, different ways to increase precision without compromising the sensitivity of the model will be explored. Note that, as it is crucial to detect as many events as possible, reduce the decision threshold to achieve higher precision is not an option, as doing so the sensitivity will decrease.

To start, the use of more complex machine learning techniques such as deep learning models are already under development. Such models show a good performance with an unbalanced dataset and might help to overcome the problem of low sensitivity. Furthermore, these are able to learn more complex and non-linear relations of the data. Secondly, the inclusion of similar outcomes such as transfer to the ICU or death might be explored. It is believed that the addition of other types of clinical deterioration might turn into an improvement on model performance while allowing to make a more impartial comparison with the MEWS and NEWS systems.

As presented in the literature, ECG telemetry data contains powerful predictive information right before the event, the addition of this information to the models might drive to an important improvement. In the same way, using high-frequency data for vital signs will allow the possibility to compute more complex features. Finally, the Bedmaster alarms are another type of signals that could lead to an improvement of the model. The database build for the development of this project already includes all this data, so there is no need to revisit the data collection stage but the data pre-processing. The idea of combining other types of predictors with the current ones is very attractive because is a promising area which has not been explored yet in the state of the art. As a suggestion, the implementation of independent early warning systems using each one a different type of predictor should be considered as a first step. Once there is a better understanding of how to use these predictors, a combined system can be developed.

A study using control samples from patients who never had an event, as well as validate the developed model with an external institution are required steps in order to ensure generalizability and functionality in any environment.

The last future research guidance refers to the deployment of the early warning model. While previous research areas are explored, and the performance of the developed early warning system is improved, is important to start exploring and planning the deployment of such systems in a hospital. Deployment planning will include but won't be limited to the development and installation of the alarm system in a central computer and training of the medical team to analyze if a specific alarm has to be considered as false or true and to interpret the output of the model.

To finally conclude with this work, it must be remembered that machine learning-based early warning systems is a research topic that has still many open questions to be answered. This work improved the state of the art by proposing an early warning system based on electronic health records and giving some insights about where to find the predictive information. It is for sure, that machine learning will play an important role in health care in the nearly future.

References

[1] **Y. Numata; M. Schulzer; R. Wal; J. Globerman; P. Semeniuk; E. Balka; J. M. Fitzgerald**. Nurse staffing levels and hospital mortality in critical care settings: Literature review and meta-analysis. In: *Journal of advanced nursing* 55 (Sept. 2006), pp. 435–48. DOI: 10.1111/j.1365-2648.2006.03941.x.

[2] **D. Wallace; D. Angus; A. Barnato; A. Kramer; J. Kahn**. Nighttime Intensivist Staffing and Mortality among Critically Ill Patients. In: *The New England journal of medicine* 366 (May 2012), pp. 2093–101. DOI: 10.1056/NEJMsa1201918.

[3] **M. Smith; J. Higgs; E. Ellis**. Factors influencing clinical decision making. In: Jan. 2008, pp. 89–100.

[4] **E. Iverson; A. Celious; C. R. Kennedy; E. Shehane; A. Eastman; V. Warren; B. D. Freeman**. Factors affecting stress experienced by surrogate decision makers for critically ill patients: Implications for nursing practice. In: *Intensive and Critical Care Nursing* 30.2 (2014), pp. 77–85. DOI: 10.1016/j.iccn.2013.08.008.

[5] **S. Virani; A. Alonso; E. Benjamin; M. Bittencourt; C. Callaway; A. Carson; A. Chamberlain; A. Chang; S. Cheng; F. Delling; L. Djoussé; M. Elkind; J. Ferguson; M. Fornage; S. Khan; B. Kissela; K. Knutson; T. Kwan; D. Lackland; C. Tsao**. Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association. In: *Circulation* 141 (Jan. 2020). DOI: 10.1161/CIR.0000000000000757.

[6] **M. Borowski; M. Görges; R. Fried; O. Such; C. Wrede; M. Imhoff**. Medical device alarms. In: *Biomedizinische Technik. Biomedical engineering* 56 (Mar. 2011), pp. 73–83. DOI: 10.1515/BMT.2011.005.

[7] **A. A. Kramer; F. Sebat; M. Lissauer**. A review of early warning systems for prompt detection of patients at risk for clinical decline. In: *Journal of Trauma and Acute Care Surgery* 87.1S (July 2019), S67–S73. DOI: 10.1097/TA.0000000000002197.

[8] **I. Wheeler; C. Price; A. Sitch; P. Banda; J. Kellett; M. Nyirenda; J. Rylance**. Early Warning Scores Generated in Developed Healthcare Settings Are Not Sufficient at Predicting Early Mortality in Blantyre, Malawi: A Prospective Cohort Study. In: *PloS one* 8 (Mar. 2013), e59830. DOI: 10.1371/journal.pone.0059830.

[9] **A. D. Bedoya; M. E. Clement; M. Phelan; R. C. Steorts; C. O'Brien; B. A. Goldstein**. Minimal Impact of Implemented Early Warning Score and Best Practice Alert for Patient Deterioration. In: *Critical Care Medicine* 47.1 (Jan. 2019), pp. 49–55. DOI: 10.1097/CCM.0000000000003439.

[10] **A. Rajkomar; J. Dean; I. Kohane**. Machine Learning in Medicine. In: *New England Journal of Medicine* 380.14 (2019). PMID: 30943338, pp. 1347–1358. DOI: 10.1056/NEJMra1814259. eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMra1814259.

[11] **A. Meyer; D. Zverinski; B. Pfahringer; J. Kempfert; T. Kuehne; S. H. Sündermann; C. Stamm; T. Hofmann; V. Falk; C. Eickhoff**. Machine learning for real-time prediction of complications in critical care: a retrospective study. In: *The Lancet Respiratory Medicine* 6.12 (Sept. 2018), pp. 905–914. DOI: 10.1016/S2213-2600(18)30300-X.

[12] **S. Hyland; M. Faltys; M. Hüser; X. Lyu; T. Gumbsch; C. Esteban; C. Bock; M. Horn; M. Moor; B. Rieck; M. Zimmermann; D. Bodenham; K. Borgwardt; G. Rätsch; T. Merz**. Early prediction of circulatory failure in the intensive care unit using machine learning. In: *Nature Medicine* 26.03 (Mar. 2020), pp. 364–373. DOI: 10.1038/s41591-020-0789-4.

[13] **M. M. Churpek; T. C. Yuen; C. Winslow; D. O. Meltzer; M. W. Kattan; D. P. Edelson**. Multicenter Comparison of Machine Learning Methods and Conventional Regression for Predicting Clinical Deterioration on the Wards. In: *Critical care medicine* 44.2 (Feb. 2016), pp. 368–374. DOI: 10.1097/CCM.0000000000001571.

[14] **J.-M. Kwon; Y. Lee; Y. Lee; S. Lee; J. Park**. An Algorithm Based on Deep Learning for Predicting In-Hospital Cardiac Arrest. In: *Journal of the American Heart Association* 7.13 (July 2018), e008678. DOI: 10.1161/JAHA.118.008678.

[15] **J. Kim; M. Chae; H.-J. Chang; Y.-A. Kim; E. Park**. Predicting Cardiac Arrest and Respiratory Failure Using Feasible Artificial Intelligence with Simple Trajectories of Patient Data. In: *Journal of Clinical Medicine* 8.9 (Aug. 2019). DOI: 10.3390/jcm8091336.

[16] **C. Shappell; A. Snyder; D. P. Edelson; M. M. Churpek**. Predictors of In-Hospital Mortality After Rapid Response Team Calls in a 274 Hospital Nationwide Sample. In: *Critical Care Medicine* 46.7 (July 2018), pp. 1041–1048. DOI: 10.1097/CCM.0000000000002926.

[17] **D. H. Do; A. Kuo; E. S. Lee; D. Mortara; D. Elashoff; X. Hu; N. G. Boyle**. Usefulness of Trends in Continuous Electrocardiographic Telemetry Monitoring to Predict In-Hospital Cardiac Arrest. In: *The American Journal of Cardiology* 124.7 (July 2019), pp. 1149–1158. DOI: 10.1016/j.amjcard.2019.06.032.

[18] **D. H. Do; J. Hayase; R. D. Tiecher; Y. Bai; X. Hu; N. G. Boyle**. ECG changes on continuous telemetry preceding in-hospital cardiac arrests. In: *Journal of Electrocardiology* 48.6 (Dec. 2015), pp. 1062–1068. DOI: 10.1016/j.jelectrocard.2015.08.001.

[19] **R. Xiao; J. King; A. Villaroman; D. H. Do; N. G. Boyle; X. Hu**. Predict In-Hospital Code Blue Events using Monitor Alarms through Deep Learning Approach. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Oct. 2018, pp. 3717–3720. DOI: 10.1109/EMBC.2018.8513269.

[20] **G. J. Escobar; V. X. Liu; A. Schuler; B. Lawson; J. D. Greene; P. Kipnis**. Automated Identification of Adults at Risk for In-Hospital Clinical Deterioration. In: *New England Journal of Medicine* 383.20 (2020). PMID: 33176085, pp. 1951–1960. DOI: 10.1056/NEJMsa2001090.

[21] **Scrum.org**. *The Scrum Framework Poster*. 2021. URL: https://www.scrum.org/resources/scrum-framework-poster (visited on 04/13/2021).

[22] **C. R. Harris; K. J. Millman; S. J. van der Walt; R. Gommers; P. Virtanen; D. Cournapeau; E. Wieser; J. Taylor; S. Berg; N. J. Smith; R. Kern; M. Picus; S. Hoyer; M. H. van Kerkwijk; M. Brett; A. Haldane; J. F. del R'ıo; M. Wiebe; P. Peterson; P. G'erard-Marchant; K. Sheppard; T. Reddy; W. Weckesser; H. Abbasi; C. Gohlke; T. E. Oliphant**. Array programming with NumPy. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[23] **W. McKinney**. Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference*. Ed. by S. van der Walt; J. Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[24] **J. D. Hunter**. Matplotlib: A 2D graphics environment. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[25] **M. L. Waskom**. seaborn: statistical data visualization. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[26] **A. Collette**. Python and HDF5. O'Reilly, 2013.

[27] **F. Pedregosa; G. Varoquaux; A. Gramfort; V. Michel; B. Thirion; O. Grisel; M. Blondel; P. Prettenhofer; R. Weiss; V. Dubourg; J. Vanderplas; A. Passos; D. Cournapeau; M. Brucher; M. Perrot; E. Duchesnay**. Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[28] **T. Chen; C. Guestrin**. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[29] **J. P. Jacobs; D. M. Shahian; R. L. Prager; F. H. Edwards; D. McDonald; J. M. Han; R. S. D'Agostino; M. L. Jacobs; B. D. Kozower; V. Badhwar; V. H. Thourani; H. A. Gaissert; F. G. Fernandez; C. Wright; J. I. Fann; G. Paone; J. A. Sanchez; J. C. Cleveland; J. M. Brennan; R. S. Dokholyan; S. M. O'Brien; E. D. Peterson; F. L. Grover; G. A. Patterson**. Introduction to the STS National Database Series: Outcomes Analysis, Quality Improvement, and Patient Safety. In: *The Annals of Thoracic Surgery* 100.6 (2015), pp. 1992–2000. DOI: 10.1016/j.athoracsur.2015.10.060.

[30] **Epic**. *User web*. 2021. URL: https://www.epic.com/ (visited on 03/15/2021).

[31] **C. Kelly; A. Upex; D. Bateman**. Comparison of consciousness level assessment in the poisoned patient using the alert/verbal/painful/unresponsive scale and the Glasgow Coma Scale. In: *Annals of emergency medicine* 44.2 (Sept. 2004), pp. 108–13. DOI: 10.1016/j.annemergmed.2004.03.028.

[32] **Z. Lipton; D. Kale; R. Wetzel**. Modeling Missing Data in Clinical Time Series with RNN. In: *Proceedings of Machine Learning Research* 56 (June 2016).

[33] **F. E. Harrell**. Regression Modeling Strategies: with applications to linear models, logistic regression, and survival analysis. New York: Springer, 2001.

[34] **L. Breiman**. Random Forests. In: *Machine Learning* 45 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[35] **T. Chen; C. Guestrin**. XGBoost: A Scalable Tree Boosting System. In: KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[36] **S. Nembrini; I. R. König; M. N. Wright**. The revival of the Gini importance? In: *Bioinformatics* 34.21 (May 2018), pp. 3711–3718. DOI: 10.1093/bioinformatics/bty373.

[37] **S. M. Lundberg; S.-I. Lee**. A Unified Approach to Interpreting Model Predictions. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon; U. V. Luxburg; S. Bengio; H. Wallach; R. Fergus; S. Vishwanathan; R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774.

[38] **C. Fernandes; S. Miles; C. J. P. Lucena**. Detecting False Alarms by Analyzing Alarm-Context Information: Algorithm Development and Validation. In: *JMIR Med Inform* 8.5 (May 2020), e15407. DOI: 10.2196/15407.

[39] **S. P. Shashikumar; G. Wardi; P. Paul; M. Carlile; L. N. Brenner; K. A. Hibbert; C. M. North; S. S. Mukerji; G. K. Robbins; Y.-P. Shao; M. B. Westover; S. Nemati; A. Malhotra**. Development and Prospective Validation of a Deep Learning Algorithm for Predicting Need for Mechanical Ventilation. In: *Chest* (2020). DOI: 10.1016/j.chest.2020.12.009.

[40] **B. Nunes; J. Pais; R. Garcia; Z. Magalhães; C. Granja; M. Silva**. Cardiac arrest: long-term cognitive and imaging analysis. In: *Resuscitation* 57.3 (2003), pp. 287–297. DOI: 10.1016/S0300-9572(03)00033-9.

[41] **J. He; S. Baxter; J. Xu; J. Xu; X. Zhou; K. Zhang**. The practical implementation of artificial intelligence technologies in medicine. In: *Nature Medicine* 25 (Jan. 2019). DOI: 10.1038/s41591-018-0307-0.

[42] **D. of the Treasury: Internal Revenue Service**. *How To Depreciate Property*. 2021. URL: https://www.irs.gov/pub/irs-pdf/p946.pdf (visited on 03/15/2021).