ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DHEST
Department of Health Sciences and Technology

Bachelor Thesis

Predictive Power of Common Risk Factors for Cardiac Arrhythmias in Critical Care

Spring Term 2019

Supervised by:
Patrick Schwab
Gaetano Claudio Scebba
Prof. Dr. Walter Karlen

Author:
Georgia Channing

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

PREDICTIVE POWER OF COMMON RISK FACTORS FOR CARDIAC ARRHYTHMIAS IN CRITICAL CARE

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Channing	Georgia

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 14/06/2019	Georgia W. Channing

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

Abstract

Cardiovascular diseases are the most common cause of natural death in developed countries with more than 450,000 fatalities annually in the United States alone. Today, cardiac arrhythmias, a group of conditions related to irregular heartbeats, are typically diagnosed based on patient-reported, qualitative symptoms. In this work, we strive to find a quantitative basis for more reliable diagnoses by attempting to measure the importance of specific factors potentially relevant to the diagnosis of cardiac arrhythmias. Using Random Forest in conjunction with data from the MIMIC-III database, we made predictions about patient diagnosis and measured feature importance. We collected data on intensive care unit (ICU) patients over the age of 16 and tested the importance of respiratory rate, blood pressure, sodium, potassium, calcium, among other features. The influence of the quantity of data was also measured by adjusting the amount of time over which data was collected. The model achieved, at its best, an Area Under the Receiver Operator Curve (AUC) score of 0.9787 and, thus, confirmed the importance of several previously suggested factors in the diagnosis of cardiac arrhythmias.

Chapter 1

Introduction

Despite recent advances in patient diagnostic procedures and medical technology, cardiovascular diseases remain the most common cause of natural death in developed countries [1]. In the United States alone, more than 450,000 people die of cardiac arrhythmias annually [2].

Cardiac arrhythmias are heart rhythm problems that occur when the electrical impulses that coordinate one's heartbeats do not work properly, causing one's heart to beat too fast, too slow or irregularly [3].

Traditionally, cardiac arrhythmias have predominantly been diagnosed based on qualitative evaluations and patient-reported symptoms. However, these are largely subjective and not always effective as they rely strongly on a patient's ability to communicate clearly with their medical professional. According to the Cleveland Clinic, more than half of sudden cardiac arrests resulting from cardiac arrhythmias occur without prior symptoms, such as chest pain, dizziness, and noticeable heart palpitations [3].

In the past several years, medical professionals have proposed experimental and quantitative biosignals as other potential markers of cardiac arrhythmias. Even Apple^{TM} has attempted to break into this field. In it's most recent attempt to use the Apple Watch to predict arrhythmia, a total of 419,297 people's heart rhythms were tracked, of which 2,161 participants (0.52 percent of the group), received arrhythmic flags [4]. However, Apple's attempts have been severely limited by it's reliance on self-reported data, unreliable sensors– and a resulting high number of false positives [4].

False positives represent a major hurdle to successfully integrating machine-generated diagnoses into regular clinical use. False positives, or the misdiagnosis of a healthy patient, may result in treatment that is harmful to a healthy patient. Although machine learning algorithms are not ready to take the place of human doctors, they can still shed light on the causes of many diseases, in particular cardiac arrhythmias. Medical professionals have only been able to conjecture that certain biosignals may be predictive of an imminent cardiac arrhythmia; machine learning algorithms can support these claims with statistical data and the identification of patterns that are not apparent to the human observer.

The goal of this work is to confirm the predictive power of various biosignals, medications, preconditions, and congenital defects for the diagnosis of cardiac arrhythmias with machine learning.

1.1 Related Work

Multiple researchers have used machine learning algorithms, including recurrent neural networks and multi-task Gaussian process models, to predict a variety of patient information, including heart failure onset, patient mortality, and future prescriptions. Their work has paved the way for the work in this thesis by (1) showing the value of machine learning algorithms in the medical field, (2) by demonstrating that specific machine learning algorithms can be suited to specific data sets, and (3) by suggesting that including temporal and trend data improves models. Their results strongly influenced the design of this project and also inspired us to use a machine learning algorithm, Random Forest, that was particularly suited to our data set.

Choi et al. [5] used recurrent neural network models to detect heart failure onset[5]. They used data from 3,884 heart failure cases against 28,903 control cases from May 16, 2000 to May 23, 2013. They adapted a recurrent neural network model to detect relationships between time-stamped diagnoses, prescriptions, and procedures for all cases [5]. They concluded that the use of time-stamped data improved the performance of deep-learning models for the early detection of heart failure during an observation window of 12 to 18 months [5].

Choi et al. [6] later developed *DoctorAI* [6]. *DoctorAI* used recurrent neural networks and time stamped electronic heath records from 260,000 patients and 2,128 physicians over 8 years to predict the diagnoses and medications for a subsequent visit based on the treatments administered during previous visits [6]. The data used for *DoctorAI* was extracted from the Medical Information Mart for Intensive Care (MIMIC) database, which can be read about more extensively in Section 2.3. They achieved 79.58 recall and also demonstrated *DoctorAI*'s adaptability by testing the model on another institution's database without losing substantial accuracy [6]. They noted that a limitation of *DoctorAI* is that its incorrect diagnoses, or false positives, can be severely damaging to a healthy patient's health if acted upon and *DoctorAI* therefore should not be used without human supervision [6].

Ghassemi et al. [7] used the MIMIC database to collect "noisy, incomplete, sparse, heterogeneous and unevenly-sampled clinical data, including both physiological signals and clinical notes" [7]. They used multi-task Gaussian process models to assess and predict patient acuity, which is the measurement of the intensity of nursing care that a patient requires [7]. The majority of acuity scores rely on a single moment of the patient's data and do not include developing clinical data such as doctors' notes, chart events or lab values [7]. They strove to refine a patient's acuity score by incorporating developing clinical data into the prediction. They first attempted to estimate cerebrovascular pressure reactivity, which often indicates the extent of secondary brain damage, such as cerebral edema or altered cerebral blood flow, in traumatic brain injury patients[7]. They also attempted to use clinical progress notes to predict patient mortality[7]. They concluded that, despite its great computation cost, this approach is effective and with some enhancements could be used regularly in a clinical setting [7].

Churpek et al. [8] studied how the addition of a patient's vital sign trends to the patient's previously-measured momentary vital signs affected the model's predictive power [8]. They used data on vital sign trends from five hospitals over a five-year period to predict cardiac arrest, hospital transfer, and death [8]. They ultimately determined that including trends increased accuracy compared to a model containing only momentary vital signs (Area under the Curve (AUC) scores of: 0.78 vs. 0.74) [8].

Schwab et al. [9] utilized a data set of more than 12,000 electrocardiogram recordings to develop a state-of-the-art ensemble of recurrent neural networks that could differentiate between normal sinus rhythms, atrial fibrillation, other types of arrhythmias and signals [9]. They extracted the time since the last heartbeat, the relative wavelet energy over five frequency bands, the total wavelet energy over those frequency bands, the R amplitude, the Q amplitude, QRS duration and wavelet entropy in order to achieve this goal [9]. They achieved results with an average F1 score of 0.79 on the private test set of the PhysioNet CinC Challenge 2017[9].

Ghassemi et al. [10] examined the use of latent variable models, which relate a set of observable variables to inferred ones, to translate hospital notes into meaningful features, and the predictive power of these features of patient's mortality [10]. Hospital mortality predictions are typically based on gender, age, scores which measure the severity of disease for patients admitted to intensive care units and scores which predict ICU mortality based on lab results and clinical data. In this work, Ghassemi et al. [10] attempted to enhance previously-made predictions by adding text information extracted from the MIMIC database [10]. They concluded that their models could ultimately be well-integrated into a clinical setting [11].

Chapter 2

Medical Background

In this section, we dive deeper into the medical background and the possible predictors of cardiac arrhythmias upon which this work is based.

2.1 Cardiac Arrhythmia

There a four primary categories of cardiac arrhythmia: premature beats, supraventricular arrhythmias, ventricular arrhythmias, and bradyarrhythmias [12, 13, 14]. Each of these results from a dysfunction of the heart's electrical impulses, or cardiac action potentials [13, 14].

Premature beats are the most common type of arrhythmia and they are typically harmless [12]. Patient-reported symptoms, though uncommon, include fluttering in the chest and feelings of a skipped beat. Premature beats typically require no treatment, especially in healthy people [12].

Supraventricular arrhythmias are tachycardias that start in the atria or the atrioventricular (AV) node [15]. The AV node is a group of cells located between the atria and the ventricles. Tachycardia is a condition that makes one's heart beat more than 100 times per minute [15]. This happens when the electrical signals in the organ's upper chambers misfire and cause the heart rate to speed up. It then beats so fast that it does not fill with blood before it contracts. Supraventricular arrhythmias require medical attention, but are not typically fatal [12, 14].

Bradyarrhythmias are arrhythmias in which the heart rate is slower than normal. If the heart rate is too slow, not enough blood reaches the brain and results in loss of consciousness [15]. Bradyarrhythmias can be caused by heart attacks, conditions that harm or change the heart's electrical activity, such as an underactive thyroid gland, aging, or an imbalance of chemicals or other substances, such as potassium or beta blockers [15]. Bradyarrhythmias require medical attention, but are not typically fatal [12, 15].

Ventricular arrhythmias, such as ventricular tachycardia and ventricular fibrillation, start in the ventricles[14]. Coronary heart disease, heart attack, weakened heart muscle, and other problems can cause ventricular arrhythmias. Ventricular arrhythmias require immediate medical attention and are often fatal [14].

The previously mentioned electrical impulse, or cardiac action potential, that governs a heart's beating is a change in voltage across the cell membranes of the heart's

cells. The cardiac action potential encompasses a series of events that produce a change in voltage the heart's cells. The change in voltage is a result of charged atoms moving between the inside and outside of the cell, through proteins called ion channels [12].

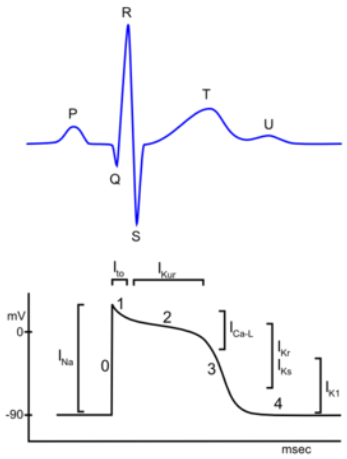

Figure 2.1: Cardiac Action Potential [16]

Though the entire process lasts only about 300 milliseconds, there are five phases of the cardiac action potential, as depicted in Figure 2.1 [12]:

0. Potassium levels decrease and fast sodium channels open resulting in rapid depolarization
1. Fast sodium channels close and sodium levels decreasing, causing partial repolarization
2. Repolarization continues as calcium ions leave the cell (plateau phase)
3. Sodium and calcium channels close, causing membrane potentials to return to baseline
4. Rest state: Adenosine triphosphatases (ATPases) [1] break down phosphate bonds, which releases energy that ATPase use to drive other cellular or chemical reactions. As three sodium ions convert to just two potassium ions, a negative intracellular potential is created.

During the vast majority of the process described above, the cell is resistant to other stimulation; another action potential will not occur until the whole previous process is complete and the cell is repolarized. If a supramaximal stimulus, a stimulus with strength significantly above that required to initiate another action potential, occurs before the previous action potential is complete, the resulting action potential is stunted [12].

The cardiac action potential drives the proper functioning of the heart and its impairment can have far-reaching effects.

2.2 Important Factors for Cardiac Arrhythmias

Because the proper functioning of the heart depends on the exact balance and timing of sodium-, calcium-, and potassium-ion channel function, measurements related to those may be a potential avenue to find more reliable, measurable predictors. As mentioned in Section 1, most cardiac arrhythmias today go undiagnosed before a cardiac event because diagnosis depends almost entirely on a patient's ability to recognize and communicate their own symptoms. A reliable and quantitative predictor of cardiac arrhythmia could represent a significant improvement in patient diagnosis.

[1] Adenosine triphosphatases (ATPases) are a group of enzymes that decompose adenosine triphosphatases (ATP) into adenosine diphosphate (ADP) and an extra phospate bond (or vice versa)[17].

Several groups of researchers have highlighted some of the lesser-known cardiac arrhythmia symptoms and trends, multiple of which are discussed Sections 2.2.1, 2.2.2, and 2.2.3. Additionally, the Clinical Guidelines for Patients with Cardiac Arrhythmias provide useful insight into what triggers a cardiac arrhythmia and sudden cardiac death [15, 14]. The features discussed in Sections 2.2.1, 2.2.2, and 2.2.3 were later tested.

2.2.1 Potassium, Calcium, and Sodium

The flow of calcium, potassium, and sodium through ion channels is integral to heart function. Multiple gene mutations, medicines, and acquired diseases are known that cause disruptions of this process [13].

Dysfunction in the heart's potassium ion channels is one cause of cardiac arrhythmia. The human ether-a-go-go (hERG) gene codes for a protein called $K_v11.1$, which is a sub-unit of the potassium ion channel [18]. This protein is one of two primarily responsible for the conclusion of the plateau phase of cardiac action potential [13]. As shown in Figure 2.2, the prolongation of the plateau phase is a sign of long-QT syndrome, which greatly increases a patient's risk of ventricular fibrillation [18]. Delayed repolarization also increases a patient's risk of Torsades de Pointes (TdP), which can also devolve into a lethal ventricular fibrillation. Mutations of other potassium-related genes such as KVLQT1, KCNQ1, and minK are also known to cause disruptions of the cardiac action potential by prolonging the plateau phase [18]. Long-QT syndrome is estimated to affect approximately 1 in 5,000 to 10,000 people worldwide and can be both hereditary and acquired [18]. There are multiple known causes of acquired long-QT syndrome, many of which are long-term, age-related diseases, but a metabolic abnormality, such as reduced serum potassium concentration (potassium deficit in the blood), is the one of the most common [18].

Dysfunction in the heart's calcium channels can also cause cardiac arrhythmia. However, these calcium channels are distributed unevenly throughout the heart and, as a result, channel dysfunction can have varying effects [13]. One possible result is a prolonged action potential, which makes the heart muscle cells particularly resistant to re-stimulation. Dispersion of this resistance can result in a unidirectional block of electrical excitement, meaning that some muscle cells will be resistant to beginning the next heart beat[13]. If an arrhythmia is triggered while the heart is in this state, the arrhythmia continues through a regenerative circuit of electrical activity around the relatively inexcitable tissue. This cycle, known as reentry, can quickly cause ventricular fibrillation and sudden death [13].

One possible cause of calcium channel dysfunction is a mutation of the L-type calcium channel CaV1.2 [19]. This protein is necessary for the correct coupling of excitation and contraction in the heart [19]. Mutations of the CACNA1C gene, which encodes the CaV1.2 protein, can also have effects in the brain, smooth muscle, immune system, teeth, and testis [19]. CaV1.2 dysfunction has been shown to cause Timothy Syndrome, a disease characterized by the abnormal union on one's fingers and toes and life-threatening cardiac arrhythmias [19]. Reduced CaV1.2 channel function, like the hERG mutation, causes prolongation of the plateau phase of the cardiac action potential and results in long-QT syndrome [19].

The SCN5A gene, responsible for the encoding of the sodium channels that initiate cardiac action potential, is also implicated as a potential cause of cardiac arrhythmia [2]. More than two decades ago, the first SCN5A mutation was discovered in multiple families with hereditary long-QT syndrome. The same mutation was later found

also in patients suffering from Brugada Syndrome, which expresses itself in specific electrocardiogram abnormalities and sudden death [11]. Since then, more than 150 SCN5A mutations have been reported, some of which are thought to cause not only long-QT syndrome and Brugada Syndrome but also cardiac conduction defect, sick sinus syndrome, atrial standstill, and susceptibility to dilated cardiomyopathy and atrial fibrillation [11].

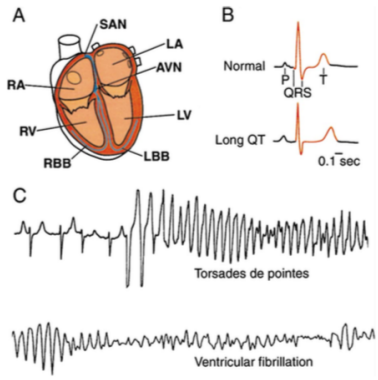

Figure 2.2: Figure 2.2 depicts the electrocardiograms (ECG) of multiple life-threatening arrhythmias that can be caused by one or a combination of the factors described in this section. A: A representation of a four-chambered heart and conducting tissues. B: A normal ECG and one showing long-QT syndrome. C: Normal breathing rhythms transforming into Torsades de Pointes ventricular tachycardia and ventricular fibrillation. Sourced from Keating and Sanguinetti [13].

Mutations of the SCN5A gene can cause both "loss-of-function" and "gain-of-function" effects, both of which negatively impact the heart's precise functioning. "Gain-of-function" mutations can cause prolongation of cardiac action potential and increase risk of long-QT syndrome, while "loss-of-function" mutations can cause shorten the cardiac action potential and higher risk of hypertension [11, 13].

Furthermore, in the failing heart, an increased number of sodium channels fail to become deactivated and continue with an inward sodium current during what should be the plateau phase [11]. This has an effect similar to that of the "gain-of-function" SCN5A mutation [11].

In summary, the successful heartbeat requires the correct balance and timing of potassium, sodium, and calcium inflow and outflow. Mutations of any of the three channels can quickly throw off the heart's delicate balance and trigger a cardiac arrhythmia.

2.2.2 Respiratory Rates and Blood Pressure

Respiratory rate is a major predictor of cardiac and pulmonary disorders and events [20]. Many cardiac arrhythmias result from hypoxia, a condition in which some part of the body is not receiving enough oxygen, and hypercarbia, a condition characterized by abnormally elevated carbon dioxide levels in the blood [21]. The human body attempts to correct hypoxia and hypercarbia by pushing more oxygen into the bloodstream [8]. This process, of breathing more and pumping more blood, elevates the respiratory rate and systolic blood pressure. Experiments conducted by Churpek et al. [8], in their attempt to foresee clinical deterioration in hospital patients, suggested that systolic blood pressure and respiratory rates are the two most predictive biosignals [8].

2.2.3 Others

As discussed in Sections 2.2.1 and 2.2.2, ion channel dysfunction and irregular cardiovascular rates can cause acquired long-QT syndrome.

Other features were included in this project, but either seemed less influential or had less extensive documentation. Many medications, including some antibiotics, antihistamines, and anti-arrhythmiacs can also cause cardiac arrhythmia [13]. Examples of such drugs include: terfenadine, cisapride, erythromycin, amiodarone, phenothiazines, tricyclic antidepressants, and certain diuretics [13]. Most of the previously listed medications facilitate long-QT syndrome by blocking hERG channels, the effects of which can be read about in Section 2.2.1. Another drug, Quinidine, induces Torsades de Pointes (TdP) in an estimated 2 to 9 percent of treated patients [18]. The recognition of the arrhythmia-inducing effects of some of these drugs has resulted in their removal from the market or relegation to restricted use, and their effects were therefore unable to be measured in this project because they are no longed prescribed in most American hospitals [18].

Finally, according to the Clinical Guidelines [14, 15], symptoms, such as syncope, dyspnea, chest pain, cardiac arrest, dyspnea and edema, should also be considered flags for cardiac arrhythmia. Doctors should also be wary of patients with histories of cocaine abuse, alcohol abuse, thyroid disease, kidney disease, lung disease, epilepsy, and hypertension [14, 15].

2.2.4 Conclusion

Multiple sources argue that multiple events are required to induce a cardiac arrhythmia and that attention should be paid to a host of biosignals in order to accurately predict a cardiac arrhythmia [13, 18, 14]. Additionally, there are more biological predictors, known and unknown, that are not considered in this project, but that may be useful in the prediction of cardiac arrhythmias.

2.3 The MIMIC-III Database

In order to measure these factors as accurately as possible, we need to train, validate, and test on very large quantities of data. We used the openly available MIMIC-III database to obtain large quantities of clinical monitoring data from patients admitted to intensive care units (ICUs).

The Medical Information Mart for Intensive Care III, or MIMIC III, "is a large, freely-available database comprising de-identified health-related data associated with over 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012" [22]. MIMIC III is primarily operated by the MIT Laboratory for Computational Physiology and associated research groups, and is notable for its large and diverse population of ICU patients, as well as detailed and complete lab test and chart event reporting.

The MIMIC III database uses patient-unique identifiers, admission-unique codes, ICD-9 and -10 codes (defined under subsection 4.1.1), as well as diagnostic codes to organize information. Any diagnosis, patient, chart event, or admission can be accessed or queried with a unique code [22].

All data referenced and used in this work were extracted from the MIMIC-III database.

Chapter 3

Technical Background

In the previous section, we discussed the medical background that underlies this paper and previous studies similar to this one. In this section, we will cover the technical and computational background 5.

3.1 Random Forest Classification

The goal of this project is to utilize the machine learning technologies developed in the last two decades to predict a patient's chance of having an arrhythmic event. There are many types of models that can be described as "machine learning", but the primary approaches relevant to this type of work fall into the categories of supervised and unsupervised learning. Supervised learning is process of a machine learning to map an input to an output based on previously given input-output pairs [23]. Later in this work, we will describe the mapping from input to output as the mapping of features to labels or classes. The goal of supervised learning is to generate an "inferred function" that maps features to labels [23]. Examples of supervised learning include Support Vector Machines, Linear Regression, Random Forest, Decision Trees, and Convolutional Neural Networks [23, 24]. The second category is called unsupervised learning, in which the dataset contains no information about what the output is [23]. More simply, unsupervised methods infer patterns from data without access to training labels [23]. The goal for these models, the most common of which is Cluster Analysis, is to reveal hidden patterns or groupings in the dataset [23].

In using the MIMIC-III database, which can be read about in more detail in Section 2.3, we had access to the model's inputs, such as patient chart events, doctor's notes, prescriptions, lab results, and outputs, arrhythmia diagnoses. Therefore, we chose to use a supervised learning method, specifically the Random Forest method.

To be precise, the problem that we attempt to solve in this work is a binary classification problem as each patient either (1) has an arrhythmia or (0) does not have an arrhythmia. There are multiple possible classifiers, or machine learning techniques that could have been used to solve this problem. We selected Random Forest because of it's accuracy and relatively quick training time [25].

3.1.1 Random Forest Algorithm

The Random Forest algorithm is used for both classification and regression. In this thesis, we will only discuss the uses of the Random Forests algorithm for classification.

As illustrated in Figure 3.1, random forests construct many individual decision trees, whose predictions are aggregated to make a final prediction. To predict a target value, decision trees split data into increasingly small subsets. Each leaf, or node, is represented by a condition or a subset of data and each branch, or edge, as a decision [26]. The splitting process continues indefinitely until a preset maximum depth is reached or no more can be learned from further splitting. When used for classification, the mode of the decision trees is taken for the final prediction.

The Random Forest algorithm is as follows [26]:

1. Given N cases, select N bootstrap samples, meaning select N cases with replacement.
2. Given M input variables, an m<<M is selected and held constant throughout the growing of the forest. At each node, rather than choosing the best split among all M variables, a random group of size m is selected and the best split from within this subset is used to split the node.
3. Each tree is grown unpruned, meaning that it is grown to the largest extent possible.
4. New predictions are made by aggregating the predictions of the previous N trees

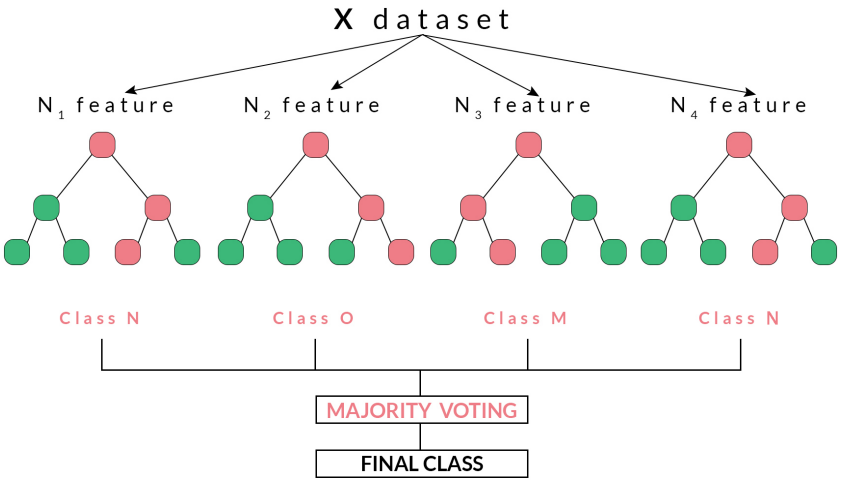

Figure 3.1: The image is a graphical represenation of the splits made by the many decision trees in the Random Forest. Image sourced from Quantinsti [27]

3.1.2 Under- and Overfitting

Underfitting refers to the case when the model, or machine learning algorithm, is unable to learn the underlying pattern, usually as a result of not having enough data. In this case, the model usually underestimates the complexity data it is being

trained on. An example of this would be a model trying to learn a linear function on non-linear data [28]. If the model is underfitted, it is also said to have "high bias" and "low variance".

Bias is the difference between the average prediction of the model and the true value [28]. Models with high bias will perform poorly on both training and test data because they have failed to understand the underlying pattern [24]. Variance is the variability of a model's prediction for a given data point. A model with high variance will likely perform well on the training set, but may not generalize well to new, unseen data. "High bias" is a result of the model relying too heavily on assumptions it made from the training data that are not necessarily part of the actual trend.

One can avoid an underfitted model by using an adequately large sample size, a proportionate number of features, and not overly limiting the depth of each tree [23, 28].

The opposite problem is called "overfitting". Overfitting occurs when the model not only learns the pattern, but also the noise in the training set. Every dataset has a certain amount of irreducible error [28], which effectively measures the amount of noise in the data. More simply, overfitting occurs when the model begins to memorize the training data. Overfitting, to which complex models such a Random Forests are especially susceptible, are symptomized by high variance and low bias [28].

Figure 3.2 depicts three models of data that roughly follow a portion of the cosine function. The leftmost image depicts a polynomial of degree 1, or linear approximation, that underfits the data. The rightmost image depicts a polynomial of degree 15 that overfits the data. Finally, the middle image depicts a polynomial of degree 4 that approximates the true function almost perfectly [24].

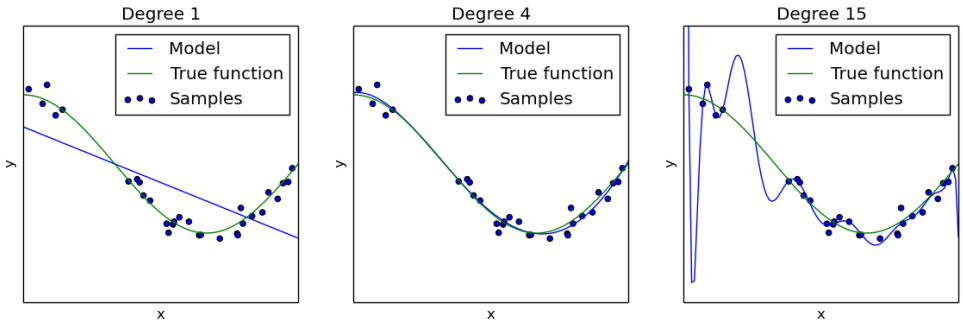

Figure 3.2: This figure shows three graphs, the leftmost is underfitted, the rightmost is overfitted, and the center is correctly fitted. Image sourced from Sci-Kit Learn [24].

Random Forests, if not limited, are prone to overfitting because the forest can continue growing until it has perfectly learned all the data, including the data's noise. In this case, the forest will grow until it has only one leaf node for every tree and has perfectly classified all of the data. While this may sound ideal, it only benefits the training set and diminishes the model's performance on the test set. Overfitting can be avoided with cross-validation, pruning, early stopping and regularization [24].

By limiting the depth of each tree, we can reduce the model's variance at the expense of increasing the model's bias [23, 29].

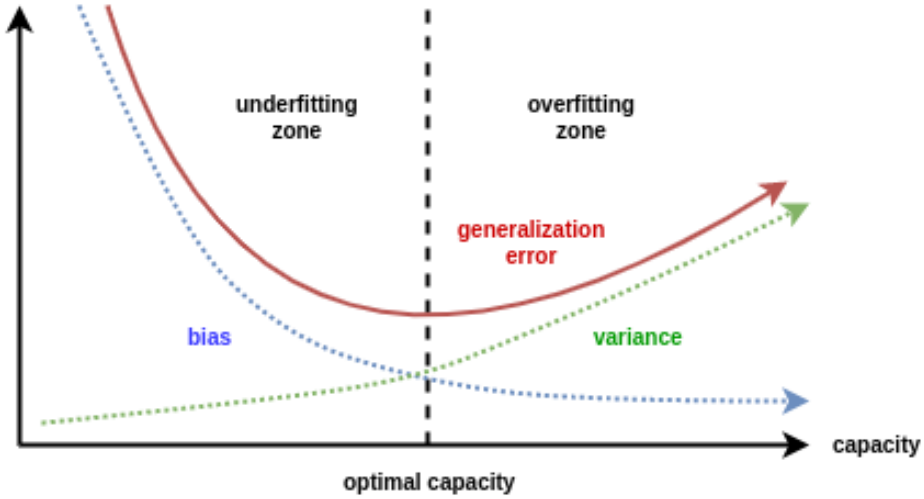

Figure 3.3: This figure depicts the trade off between Bias and Variance. Figure sourced from Towards Data Science Magazine [28].

As shown in Figure 3.3, finding the balance between bias and variance, or between under- and overfitting, is known as the Bias-Variance trade-off and is an important aspect of building a model [23, 29].

Let the true function for the data the model is learning be $f(x)$ and the function that the model extrapolates be $f'(x)$, the expected squared error at a given point x is [23]:

$$Err(x) = E[((f(x) - f'(x))^2] \tag{3.1}$$

This can be seen also as [23]:

$$Err(x) = (E[f'(x)] - f(x))^2 + E[(f'(x) - E[f'(x)])^2] + \sigma_e^2 \tag{3.2}$$

or

$$Err(x) = Bias^2 + Variance + IrreducibleError \tag{3.3}$$

3.1.3 Gini Impurity

The Gini Impurity is used to determine on which feature the model will split. Node impurity, or Gini Impurity, is the probability that a new instance would be incorrectly classified if the new instance were randomly classified by the distribution of labels in the data set[28]. The Gini Impurity represents the probability of a new data point being incorrectly classified based on the training we have already observed.

The Gini Impurity of a node n is calculated by the following formula [28, 26]:

$$I_G(n) = \sum_{i=1}^{J}(p_i) \times (1 - p_i) = 1 - \sum_{i=1}^{J}(p_i)^2 \tag{3.4}$$

where J are all classes (in a binary classification J = 2) and p_i is the probability of a classification i.

At each node, each decision tree uses the Gini Impurity to choose which feature to split the node on. The decision tree chooses the feature that most decreases the Gini Impurity for the node in question[28]. It repeats this process recursively on each node until the tree reaches its preset maximum depth or no further information can be gained. If each node contains only samples from one class, no further information can be gained [28].

At the last possible layer, the Gini Impurity goes to zero, meaning that there is no chance that a new instance would be misclassified[28].

3.1.4 Gini Importance

Random Forests allow for the calculation of feature importances using the Gini Importance measure [26]. Feature importance measures how much each feature contributes to the predictiveness of the model. The higher the number the greater the contribution. In theory, feature importance is measured by calculating the increase in the model's prediction error after permuting the feature or shuffling its values. The more the permutation of the feature increases the error, the more "important" a feature is. This should be apparent, since, if the model relies more significantly on the feature, shuffling its values will have a greater effect. More concretely, feature importance is generally measured by the mean decrease in node impurity offset by the probability of reaching the same node.

In Sci-Kit Learn, the feature importance functions uses the Gini Importance for a binary tree:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{3.5}$$

where ni_j is the node importance of node j, w_j is the number of weighted samples reaching node j, and C_j is the impurity of node j [26].

The importance of each feature is then calculated by:

$$fi_i = \frac{\sum_{\text{j: node j splits on feature i}} ni_j}{\sum_{n \in \text{all nodes}} ni_n} \tag{3.6}$$

where where fi_i is the importance of feature i and ni_j is the importance of node j. These values can be normalized and averaged over all trees to find feature importance at the forest level.

3.2 Metrics

Metrics, or scores, are used to evaluate the quality of the model's predictions. Multiple metrics are used to get a comprehensive picture of how the model performed. More specifically, we use multiple metrics to better understand the different types of error there are and how well our model predicts different aspects of the dataset.

The majority of the follwing metrics are based on the idea of a "confusion matrix". A confusion matrix is used to calculate a model's Recall and Precision Scores, which ultimately affect many of the other metrics discussed below. An example of a confusion matrix follows in the figure below:

A "True Positive", or TP, refers the model's correct prediction that a patient will be diagnosed with cardiac arrhythmia. A "False Positive", or FP, refers to the model's

True Positive	False Positive
True Negative	False Negative

Table 3.1: The Confusion Matrix is represented here simply with text, but in practice would have numbers in each cell. Specifically the number of True Positives, True Negative, False Positives, and False Negatives.

incorrect prediction that a patient will be diagnosed with cardiac arrhythmia. A "True Negative", or TN, refers to the model's correct prediction that a patient will not be diagnosed with cardiac arrhythmia. Lastly, a "False Negative", or FN, refers to the model's incorrect prediction that a patient will not be diagnosed with cardiac arrhythmia.

3.2.1 Recall

The recall score is calculated by dividing the total number of TPs by the sum of the TPs and FNs [24]. In a perfect model, a recall score would be 1.0, as there would be no false negatives. The closer the model's recall score is to 1.0, the better the model is.

3.2.2 Precision

The precision score, similar to the recall score, is calculated by dividing the total number of TPs by the sum of the TPs and FPs [24]. Again, one hopes for as few false positives as possible and aims for a precision score of 1.0.

3.2.3 Average Precision

The Average Precision score encapsulates the precision-recall curve by using the weighted mean of precisions at each threshold weighted against the increase in recall from the previous threshold [24]:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{3.7}$$

where P_n and R_n represent the precision and recall, respectively, at threshold n [24].

3.2.4 F1 Score

The F1 score, or the F measure, is useful to compare models with dissimilar precision and recall scores. The F1 score attempts to measure both precision and recall simultaneously, using the following equation [24]:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{3.8}$$

The F1 score emphasizes the outlying smaller values and suppresses outlying larger values by using the Harmonic Mean, rather than the Arithmetic Mean.

3.2.5 Hamming Loss

The Hamming Loss computes the Hamming Distance between two sample sets in order to determine was percentage of the sample was mislabelled [24]. A simple example follows:

$$y_{true} = [4, 8, 12, 16]$$
$$y_{predicted} = [4, 8, 12, 15]$$

The Hamming Loss here is 0.25; one quarter of the sample was mislabelled.

Given that p_j is the predicted value of the label j, y_j is the true value of label j, and n is the number of labels, the Hamming Loss between two samples is defined as [24]:

$$L_{Hamming}(y, p) = \frac{1}{n} \sum_{j=0}^{n-1} 1(p_j \neq y_j) \tag{3.9}$$

where $1(x)$ is an indicator function.

3.2.6 The Area Under the Curve Score

The Area Under the Curve Score, or AUC score, is a summary measurement for classification problem at various thresholds. The AUC score measures the area under the ROC curve, which stands for the Receiver Operator Characteristic Curve. The ROC curve is a probability curve and is a graphical representation of the diagnostic ability of a binary classifier [24, 30]. More simply, it depicts how well the model is able to differentiate between classes. ROC curves typically have the Recall, otherwise called the True Positive Rate (TPR) or Sensitivity, on the Y axis and the False Positive Rate (FPR), or probabilistic inverse of the specificity, on the X axis [24].

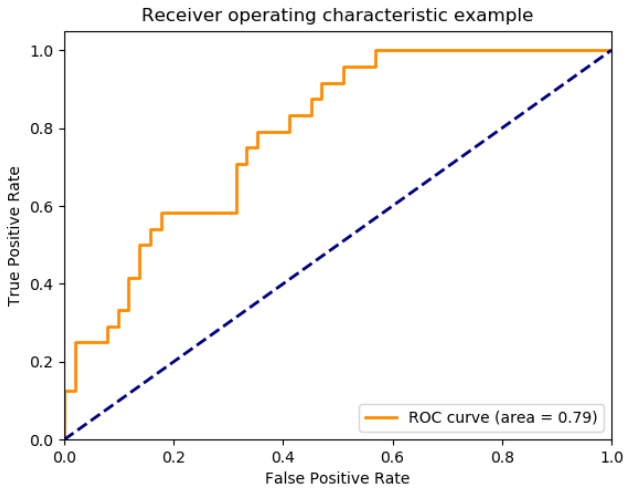

Figure 3.4: Image sourced from Sci-Kit Learn User Guide [24]. The orange ROC curve with an AUC score of 0.79 and a dotted line which represents a ROC curve with an AUC score of 0.5.

Figure 3.4 depicts an orange ROC curve with an AUC score of 0.79 and a dotted line which represents a ROC curve with an AUC score of 0.5. A perfect model would have an AUC score of exactly 1, meaning that it perfectly separated the data into the correct classes. A model with an AUC score of 0 has completely inversed the classes and maps 0s to 1s and 1s to 0s. The worst possible AUC score is that of 0.5, meaning that the model has exactly no class-differentiation abilities [24].

This means that at top left corner, or area closest to (1,0), is the "ideal" point [24]. This rating is almost impossible to achieve, but, as the "ideal" point approaches (1,0), it maximizes the area under the curve.

The Figures 3.5 and 3.6 plot the distributions of classification probabilities. Figure 3.5 shows a model that is perfectly able to distinguish between positive class and negative class [30]. When two curves have no overlap, there is no case that the model misclassified.

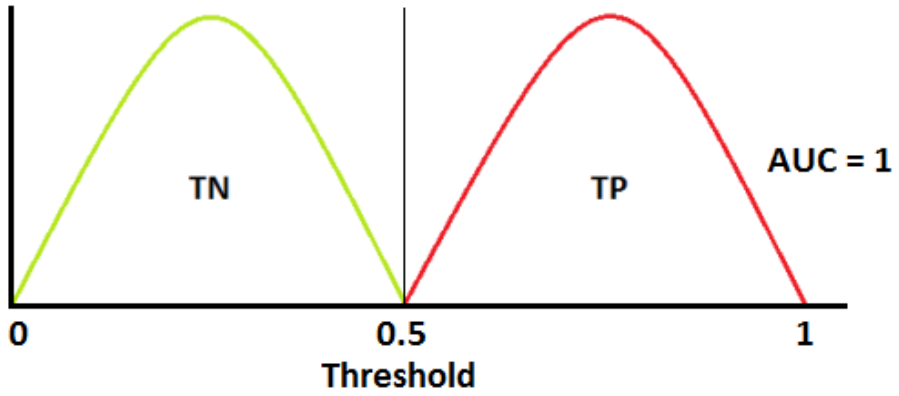

Figure 3.5: A model that is perfectly able to distinguish between positive class and negative class with an AUC of 1. Image sourced from Towards Data Science Magazine [30]

Figure 3.6 shows a model that is only able to differentiate between the positive and negative class 70 percent of the time; 30 percent of the samples cannot be reliably classified by the model.

3.2.7 Specificity, Sensitivity and Threshold in the ROC Curve

Sensitivity is another word for Recall, which again is the number of true positives divided by the sum of the true positives and false negatives. Specificity is 1 - FPR, the false positive rate. Specificity is calculated by dividing the number of true negatives by the sum of the false positives and true negatives [24, 30]. Threshold is the limit at which we classify a non-binary prediction as a Positive or Negative. By manipulating the threshold, we can minimize or maximize Type 1 Error, the number of False Positives, and Type 2 Error, the number of False Negatives [31].

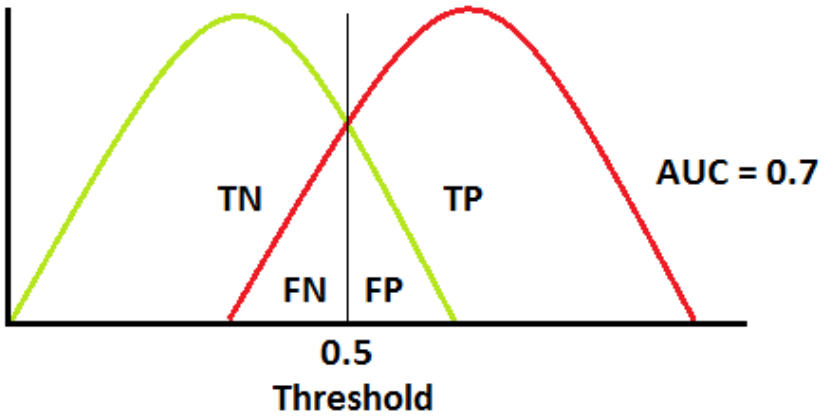

Figure 3.6: A model that is only able to differentiate between the positive and negative class 70 percent of the time. Image sourced from Towards Data Science Magazine [30]

Chapter 4

Methodology

In this Chapter, we will discuss how the the data from the MIMIC-III database was prepared for training, validation, and testing. The following flow chart in Figure 4.1 shows a simplified version of the process. Each bracketed group of processes represents a section in this chapter.

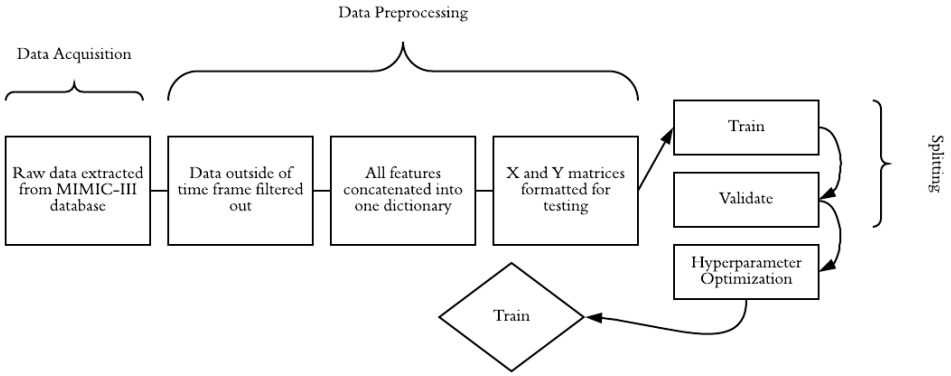

Figure 4.1: In this figure, the model pipeline is shown from raw data acquisition to testing. Each grouping, for example "Data Preprocessing", represents a subsection of this chapter.

4.1 Data Acquisition and Implementation

All the data used in this project was accessed from the MIMIC III databse and stored in a SQLite database. The data was cleaned and processed with Python 3.0 and the model was trained with a variety of Sci-Kit Learn libraries.

In order to utilize the MIMIC's by-patient-id search optimizations, we optimized SQLite queries in many functions by looping through patient identifiers, rather than creating more expansive searches. In a test done with getting the patients' admit times, a loop query completed the task in 17.9274 seconds, while the batch query needed 102.4190 seconds.

A single function, using mutable ICD-9 codes to specify the desired feature or diagnosis, was implemented for most queries. According to the Centers for Disease

Control and Prevention [32]:

"The International Classification of Diseases (ICD) is designed to promote international comparability in the collection, processing, classification, and presentation of mortality statistics. This includes providing a format for reporting causes of death on the death certificate. The reported conditions are then translated into medical codes through use of the classification structure and the selection and modification rules contained in the applicable revision of the ICD, published by the World Health Organization."

We implemented similar functions to query for specific drugs, diagnoses, and chart events.

4.1.1 SQLite

SQL stands for Structured Query Language and is the standard language for relational database management systems. SQL commands typically update databases or retrieve data from them.

SQLite is a library based in C that implements a miniature SQL that generally follows standard PostgreSQL syntax. SQLite is the most used database in the world and was founded in 2000 [33].

4.2 Data Preprocessing: Selection

In summary, the The MIMIC database was processed on the Health Sciences and Technology (HEST) Cluster, filtered to be with in a specified time frame, organized into a multi-class X and a single-class Y, and finally split into train, validation, and test sets.

Data preprocessing and cleaning comprised much of the heavy-lifting in this project. The software is configurable through program arguments and is described in more detail in the subsection Program Arguments 4.2.1.

Two time constraints were considered. The number of hours of data to be measured and which hours they should be.

In order to respect the time constraints denominated in the program arguments, data from non-binary biosignals (e.g. respiratory rates) was collected only within a specific time window. Each data point's time stamp was used to filter out data points outside the preset time frame.

Figure 4.2 depicts a flowchart that may clarify the following explanation of how cases were included or excluded from the study. The time frame was chosen differently for patients who had been diagnosed with arrhythmia and those who had not. We further differentiated between patients who had been medicated for said arrhythmia and those who had not. Despite the fact some patients had not been medicated, they were nonetheless included in some of our tests. The data of medicated patients diagnosed with cardiac arrhythmia was measured for the preset number of hours before their time of first medication. The presciption of any of the following anti-arrhythmiac medications would include a patient in the "medicated patients" group: Procainamide, Feclainide, Sotalol, Metoprolol, Toprol, Verapamil, Digoxin, Amiodarone, Potassium Chloride, Torsemide, or Sodium Chloride. The data of unmedicated patients diagnosed with cardiac arrhythmia was measured for

the preset number of hours after the admission in which they were diagnosed with cardiac arrhythmia (as a patient could have multiple distinct admissions). Patients diagnosed with cardiac arrhythmia were medicated on average within 11 hours 16 minutes and 17 seconds of admission [34]. As a result, patients diagnosed with cardiac arrhythmia but not prescribed one of the above listed medications were included in tests in which time frames were 24 or 12 hours (but not for those in which time frame was 6 hours).

The data of patients who were not diagnosed with a cardiac arrhythmia were measured for the preset number of hours after their earliest admission. Patients under the age of 16 were not included in any of the training, validation, or testing processes, as a child's normal respiratory rate is very different than that of an older patient [35, 12]. It should also be made clear that the time of medication does not necessarily denote the time of a cardiac event, but was the best way to approximate the time of a cardiac event.

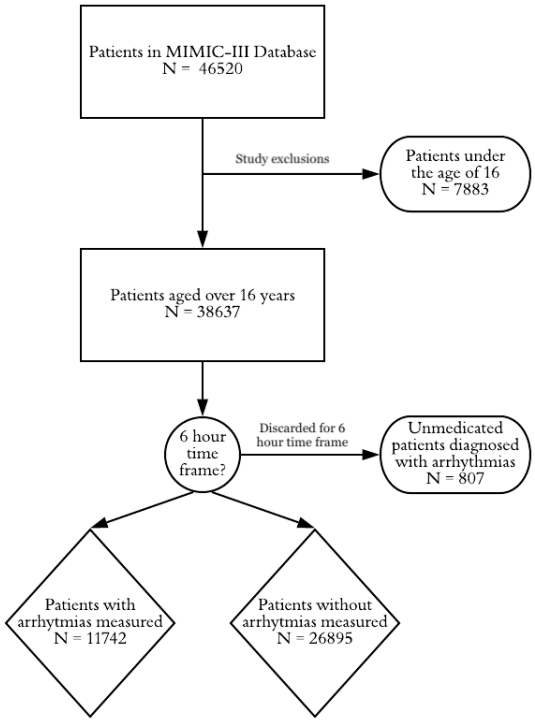

Figure 4.2: This flowchart depicts how cases were discarded from analysis.

After the number of hours was selected, each patient's lab values and chart events were filtered so that only those in the specified time frame were measured. Though each patient could potentially have multiple measurements for a single feature, only one value per feature was associated with each patient when the model was trained on the data. For each feature, we used the median of all the patient's data points over the selected time frame.

Features were not normalized because, unlike many other algorithms, outliers do not affect the Random Forests ability to make decisions. As discussed in Section 3.1.3, the Random Forest makes splits depending on what most decreases the Gini Impurity and can make those splits no matter the scale of variations in data.

4.2.1 Program Arguments

The program included a variety of run-time arguments that generally fall into one of two categories: biosignal selection or model modification.

Biosignal selection allowed the user to select any combination of the following biosignals for prediction: respiratory rates, potassium levels, sodium levels, calcium levels, and blood pressure. It also allowed the user to select histories of the following: quinine, terfenadine, astemizole, cocaine use, alcohol abuse, muscular dystrophy, renal

failure, heart failure, epilepsy, lung disease, and pulmonary circulation disorder. Biosignal selection options also included qualitative symptoms, such as dyspnea, labored breathing, and angina, chest pain.

Parameters for model modifications included choices for: database, validation set fraction, test set fraction, various output files, maximum depth of tress in the random forest, maximum number of trees in the random forest, inclusion/exclusion of child patients, the number of hours of chart events to measure, and the number of patients to be included. The number of patients included could also be offset, so that an exact range of patient identifiers could be selected. Data was collected and measured on exclusively the patient identifiers that appear in the list of "loaded patients".

4.3 Data Preprocessing: Organization

We built two data matrices for training: The y_{true} matrix, in which the true outcomes for each patient are stored, and the x_{data} matrix, in which the selected biosignals for all the loaded patients were stored.

The y_{true} matrix was a one-dimensional matrix that contained only zeros and ones, zeros for patients who have not been diagnosed with cardiac arrhythmia and ones for patients who have been diagnosed with cardiac arrhythmia. The matrix was ordered by patient identifier, so that the rows of the x and y matrices line up correctly.

The x_{data} matrix was an m by n matrix, where m was the number of biosignals measured and n was the number of patients loaded. The x and y matrices were both converted, from the Python dictionaries that originally stored the un-ordered information, with the same pre-processing function that removed the patient identifiers (as the model should not be trained on random identifiers) and converted the data type to Num-Py array.

We created the y_{true} dictionary by loading the list of patients that have been diagnosed with cardiac arrhythmia, creating a dictionary with all patient identifiers from loaded patients as keys, and assigned ones or zeros for diagnosis or not of cardiac arrhythmia.

We created the x_{data} by making, for each selected biosignal, an individual dictionary with either zeros and ones for binary biosignals or median values for non-binary biosignals. These dictionaries were concatenated per patient to make a multi-value dictionary, which was also eventually converted into a NumPy array.

4.3.1 Splits

The final step before training was the splitting stage. Splitting data was an important aspect of machine learning and is used to verify that the model generalizes well to new data. Splitting allowed us to test on data that the model has not yet seen, which in turn prevents overfitting.

In the splitting stage, the data was split into three categories: train, validate, and test. The data is split stratified to avoid a large imbalance in the distribution of the target class [24]. This could, for example, be a case in which all patients with arrhythmias were put into the validation set and, as a result, all of the training data becomes completely useless. To avoid an imbalanced distribution, we used a

Stratified Shuffle Split as seen in Figure 4.3. The Stratified Shuffle Split is a variation of the standard Shuffle Split, which generates a user-specified number of test and train splits from shuffled data [24]. Stratified Shuffle Split further guaranteed an equal distribution by creating splits with the same percentage for each target class as in the complete set [24].

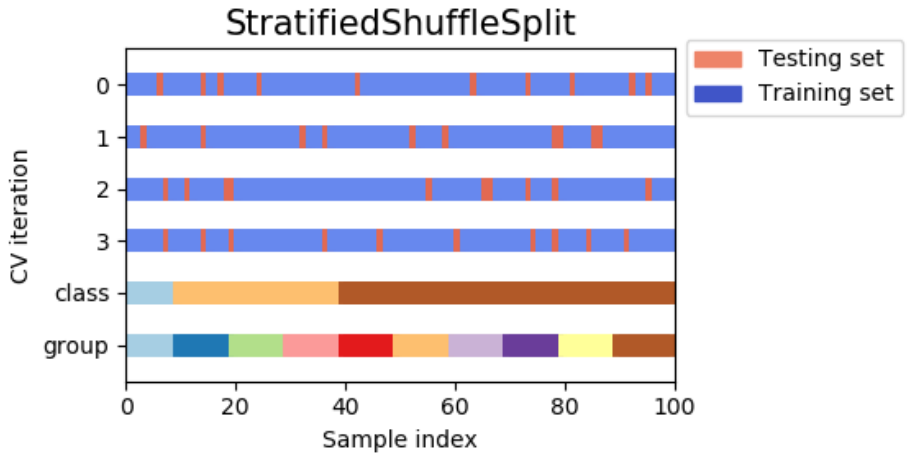

Figure 4.3: Graphic depicts a Stratified Shuffle Distribution between a test and training set. Image sourced from Sci-Kit Learn User Guide [24]

In this project, we split the data into the aforementioned three splits, train, validate, and test, with respective weights of 0.7, 0.2, and 0.1.

4.3.2 Libraries

To implement our data processing pipeline, we used the following openly available libraries.

Sci-Kit Learn

Ski-Kit Learn, formerly scikits.learn, is a free machine learning library with a Python interface, though c-libraries are sometimes used for arrays and matrix operations [36]. The library includes various tools for classification, regression and clustering and is designed to seamlessly operate with NumPy and SciPy.

The project began in 2007 as a Google Summer of Code project by David Cournapeau. Later in 2007, Matthieu Brucher incorporated the project into his thesis [36]. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel took over the project and made the first public release in February 2010 [36]. Since then, several releases have appeared following an approximately 3 month cycle, and a variety of other contributers have furthered the project [36].

Python Libraries

Python is freely usable and distributable as it is developed under an OSI-approved open source license[37]. The license is administered by the Python Software Foundation [37]. The Python Package Index (PyPI) hosts thousands of third-party modules

and libraries for Python[37]. Three of such libraries were used extensively in this project and are elaborated upon below. For this project, we used Python 2.7.

1. **Argparse**: The argparse library allows for the creation user-friendly command-line interfaces and was used extensively in *ProgramArguments* file. This program defines what arguments a program requires and parses those arguments into an easily-usable Python dictionary[37]. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments [37]. This package was used in conjunction with Python 2.7 and has therefore been updated since the writing of this paper.
2. **Datetime**: The datetime module allows users to easily manipulate dates and times in both simple and complex ways [37]. The package supports date and time arithematic, parsing, and intelligent formatting [37]. This module was used primarily for the parsing of time stamps on chart events and lab values and for manipulating patient time frames.
3. **NumPy**: NumPy, a submodule of the SciPy library, is sponsored by Enthought [38]. NumPy is the primary package used for scientific computing with Python [38]. This package contains a powerful N-dimensional array object, basic linear algebra functions, basic Fourier transforms, sophisticated random number capabilities, tools for integrating Fortran code, and tools for integrating C/C++ code [38]. The Numpy package allows for the easy conversion of Python dictionaries into C arrays, which the Sci-Kit libraries require for training.

4.4 Hyperparameter Optimization

The choice of hyperparameters, such as the maximum number of trees in the random forest, significantly affects the model's runtime during training and its performance on the validation set. The goal in optimizing hyperparameters is to find the approximately optimal values for hyperparameters, such as the maximum number and depth of trees in the forest, that will yield the best results.

In this program, the fine-tuning of hyperparameters is only done when the "split" program argument is set to "validate". When configuring multiple sets of hyperparameters, the Area Under the Curve, or AUC metric, was used to judge the quality of the tested hyperparameters.

The AUC score, which is more extensively documented in the Scores section 3.2, is briefly described by Google Developers as [39]:

"AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example."

The hyperparameters that maximized the AUC score should then used for testing. However, each model could have its own hyperparameters, of which there are also many. In order to efficiently find appropriate hyperparameters, we used sklearn's GridSearchCV [24]. Sklearn's GridSearchCV implements a cross-validated grid search of possible hyperparameters.

One aspect of hyperparameter optimization is finding the optimal number of trees in the forest and the optimal depth of each of those trees. For each model, the number and depth of trees was tested with GridSearchCV. As a rule, the greater

number of trees the more accurate the model becomes, however one should also be wary over overfitting. As seen in the chart below, the model's performance peaks at 64 trees.

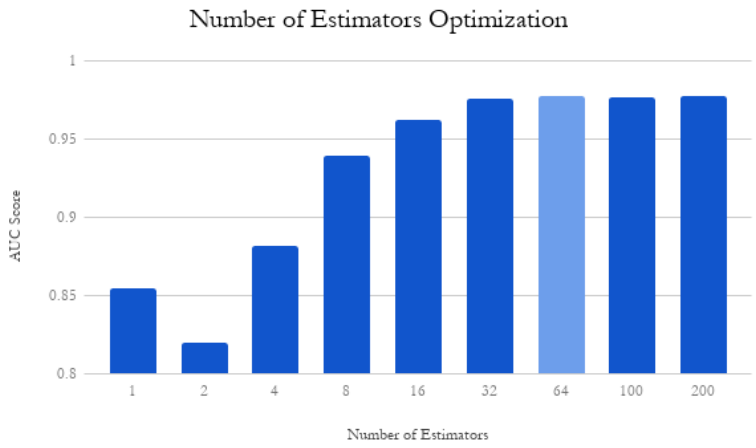

Figure 4.4: Graphic depicts results from validation run with 19 biosignals, where a maximum number of trees of 64 begets an AUC score of 0.977534294463038.

The depth of the tree determines how many splits the tree is allowed to make; this model peaks at a depth of approximately 25.

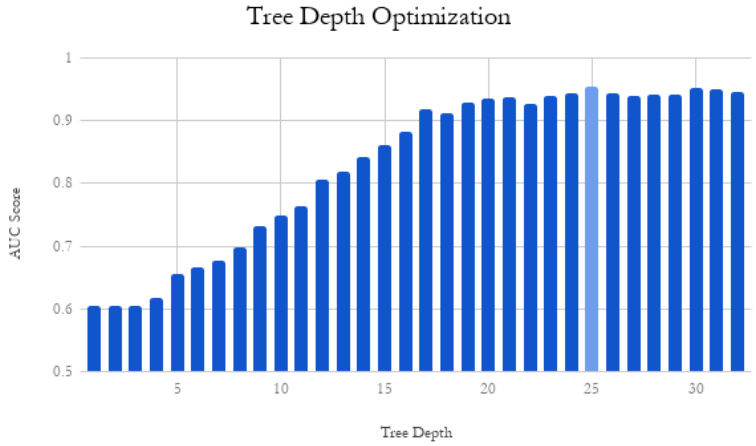

Figure 4.5: Graphic depicts results from validation run with 14 biosignals, where a maximum depth of 25 begets an AUC score of 0.95358157841925.

4.5 Testing

The finished model tested 16 biosignals, the selection of which is described in Section 2.2, and a variety of combinations of those 16. However, in order to test all possible

combinations, we would have had to do:

$$\sum_{k=1}^{n=16} \binom{n}{k} = 65,535 \tag{4.1}$$

tests. This is neither time-efficient nor useful, so instead we used the following method to determine which combinations to test.

Upon running the first test, it was discovered that some features had importances of exactly 0.0 because no patient in the database had a recording for those features and they were subsequently stripped from further tests. We ran tests with 24-, 12-, and 6- hour time frames.

An initial test was run with all features with non-zero importances for each time frame. The combination for each subsequent run was selected by systematically removing the biosignal with the highest feature importance. In repeatedly removing the most important feature, we were eventually left with a set of relatively useless features and tests were cut off after the AUC score fell under .530.

A few other tests were run to collect other data, as seen in the Results Section 5.1

Chapter 5

Analysis

In Chapters 2 and 3, the reader was equipped with the knowledge to understand the Methodology Section and the Results and Discussion which follow.

5.1 Results

The following five features, Terfenadine, Quinidine, Astemizole, Epilepsy, and Dyspnea, had feature importances of exactly 0.0 and were removed after the first test because they only confused the model with useless data [34]. The model's AUC score increased once these features were removed.

With only the five most predictive features, calcium, sodium, blood pressure, potassium, and respiratory rates, the model still achieved an AUC score 0.96797 on the 24 hour model [34].

As seen in Figures 5.1, 5.2, and 5.3, the model achieved AUC scores of 0.979, 0.962, and 0.947 with times frames of 24, 12, and 6 hours respectively when all of the features with non-zero importances were included.

As seen in Figures 5.1, 5.2, 5.3,the feature importances varied slightly depending on the length of the time frame, but generally feature importances remained within a margin of error.

As is evident from Figures 5.1, 5.2, and 5.3, the model predominantly used the same five features to split nodes: Sodium, Potassium, Calcium, Respiratory Rate, and Blood Pressure. The difference in the median values for those features between patients diagnosed with arrhythmia and those not diagnosed is graphically depicted in Figure 5.4. P-values were calculated to better understand the statistical significance of the results displayed in Figure 5.4. To better understand these values, most authors refer to statistically significant as a p-value of less than 0.05 and highly statistically significant as p-value of less than 0.001 [38]. The following two-tailed p-values were calculated from a t-test, a two-sided test for the null hypothesis that two independent samples have identical expected averages [38]. The p-values for calcium, sodium, potassium, respiratory rate, and blood pressure were 7.84×10^{-37}, 0.01, 3.53×10^{-14}, 3.47×10^{-39}, and 5.85×10^{-88} respectively [34]. The differences between all of these features were statistically significant.

To contextualize Figure 5.4, we discuss typical values for each of these features.

24hr Time Frame							
	AUC	F1	Average Precision	Recall	Accuracy	Hamming Loss	Specificity
All	0.98	0.97	0.94	0.97	0.98	0.02	0.99
Unimportant Factors*	0.98	0.97	0.95	0.97	0.98	0.02	0.99
Sodium	0.94	0.92	0.89	0.89	0.96	0.04	0.98
Potassium	0.80	0.75	0.69	0.64	0.87	0.13	0.97
Blood Pressure	0.68	0.54	0.51	0.41	0.79	0.21	0.95
Respiratory Rates	0.58	0.29	0.38	0.18	0.73	0.27	0.97
Calcium	0.54	0.19	0.34	0.11	0.71	0.29	0.96
Lung Disease	0.53	0.16	0.33	0.09	0.70	0.30	0.97
Heart Failure	0.50	0.00	0.30	0.00	0.70		

Table 5.1: This table shows the score results when the model was run with a time frame of 24 hours [34]. After each test, the most important feature in that test was removed before running a new test. For example, "Sodium" denotes that a patient's sodium level removed from the set of features from the previous line. *Terfenadine, Epilepsy, Quinidine, Astemizole, Dyspnea.

12hr Time Frame							
	AUC	F1	Average Precision	Recall	Accuracy	Hamming Loss	Specificity
All	0.96	0.95	0.91	0.95	0.97	0.03	0.97
Potassium	0.93	0.91	0.86	0.90	0.95	0.05	0.97
Sodium	0.80	0.74	0.70	0.60	0.87	0.13	0.99
Blood Pressure	0.67	0.51	0.49	0.37	0.78	0.22	0.96
Calcium	0.55	0.23	0.35	0.14	0.71	0.29	0.96
Respiratory Rates	0.51	0.06	0.31	0.03	0.70	0.30	0.99

Table 5.2: This table shows the score results when the model was run with a time frame of 12 hours [34]. After each test, the most important feature in that test was removed before running a new test. For example, "Sodium" denotes that a patient's sodium level removed from the set of features from the previous line. *Terfenadine, Epilepsy, Quinidine, Astemizole, Dyspnea.

6hr Time Frame							
	AUC	F1	Average Precision	Recall	Accuracy	Hamming Loss	Specificity
All	0.95	0.91	0.85	0.94	0.95	0.05	0.95
Potassium	0.92	0.88	0.81	0.88	0.93	0.07	0.95
Blood Pressure	0.85	0.80	0.70	0.76	0.89	0.11	0.94
Sodium	0.71	0.59	0.54	0.46	0.82	0.18	0.97
Calcium	0.5676	0.27	0.35	0.17	0.74	0.26	0.96
Respiratory Rates	0.51	0.06	0.29	0.03	0.72	0.28	0.99

Table 5.3: This table shows the score results when the model was run with a time frame of 6 hours [34]. After each test, the most important feature in that test was removed before running a new test. For example, "Sodium" denotes that a patient's sodium level removed from the set of features from the previous line. *Terfenadine, Epilepsy, Quinidine, Astemizole, Dyspnea.

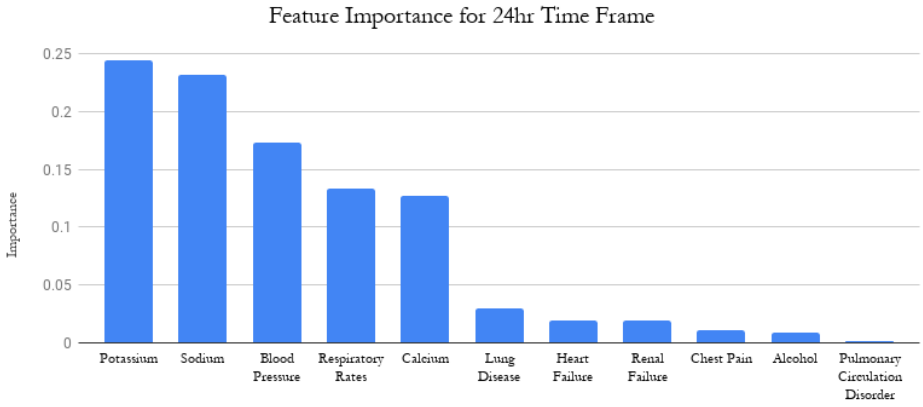

Figure 5.1: This histogram depicts the distribution of feature importance when the time frame was 24 hours.

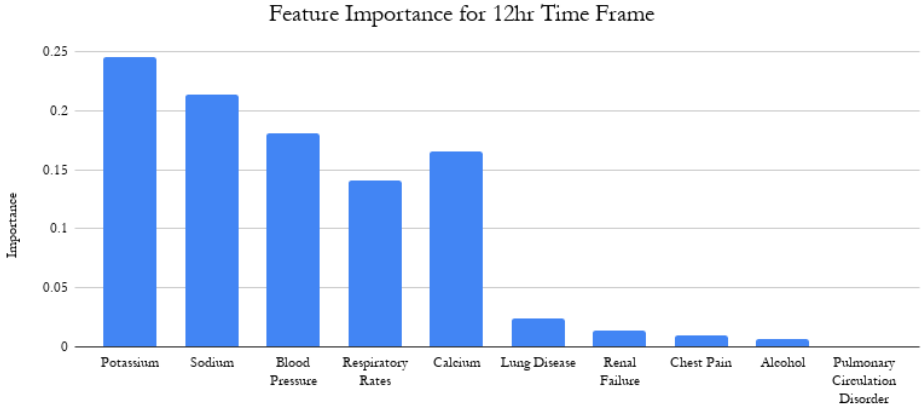

Figure 5.2: This histogram depicts the distribution of feature importance when the time frame was 12 hours.

According to the American Heart Association, a blood pressure [1] of above 80 means that a patient has Stage 1 Hypertension [35]. While the median blood pressure of a patient with arrhythmia is not over 80, it is closer than those without arrhythmias. A normal respiratory rate falls within 12 to 20 breaths per minute [35]. In adults, the healthy range of potassium is from 3.5 to 5.1 mmol/L; A serum potassium concentration greater than the upper limit of the normal range is dangerous and can quickly cause lethal cardiac arrhythmia [40]. The healthy reference ranges for serum sodium and serum calcium are from 136 to 145 mmol/L and from 8.9 to 10.1 mg/dL respectively [40]. It should be noted that the vital signs and serum values of other patients in the ICU are also likely different from that of the average healthy person.

[1] Diastolic Blood Pressure

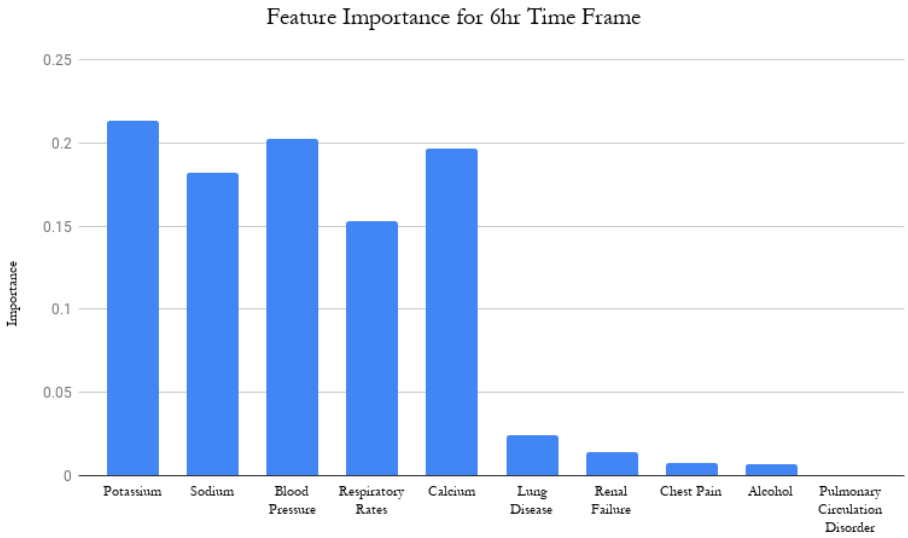

Figure 5.3: This histogram depicts the distribution of feature importance when the time frame was 6 hours.

5.2 Discussion and Limitations

The fact that Terfenadine, Quinidine, and Astemizole had no influence is likely because their potential to cause arrhythmias has long been known and they were therefore not prescribed to patients in the MIMIC database [34, 13, 18].

As is seen in Figures 5.1, 5.2, and 5.3, the top five features are significantly more predictive than any of the others. It should be noted, that the Clinical Guidelines only suggest observing features that our model determined were relatively less important [14, 15]. However, the features that were most important, Sodium, Potassium, Calcium, Blood Pressure, and Respiratory Rates, are also those that were most commonly observed in patients. The numbers of patients with data for serum potassium (34,999), serum sodium (34,843), serum calcium (13,597), respiratory rate (26,166), and blood pressure (26,146) were significantly greater the numbers of patients with with lung disease (5,200), renal failure (2139), alcohol abuse history (1224), and chest pain (1094). The greater quantity of patients with data for serum potassium, serum sodium, serum calcium, respiratory rate, and blood pressure could have skewed feature importances.

It is possible that effects of serum potassium, sodium and calcium were exaggerated by the Clinical Guidelines' recommendations to in some cases treat patients by repleting a patient's blood with these serums [14]. Specifically, for patients diagnosed with Torsades des Pointes, the Clinical Guidelines recommend using potassium or magnesium repletion to treat patients [14]. Furthermore, the Clinical Guidelines state that treating patients with Premature Ventricular Contractions with sodium channel blockers, which are used to treat some other types of arrhythmias, actually increases a patient's risk of death [14]. Calcium channel blockers are also thought to be harmful in patients with complex tachycardias and the Clinical Guidelines suggest that increasing a patient's serum calcium levels may be helpful in treating tachycardias[15]. Possible errors could have resulted from not knowing the exact time of a cardiac event. We estimated that a cardiac event was at the time or

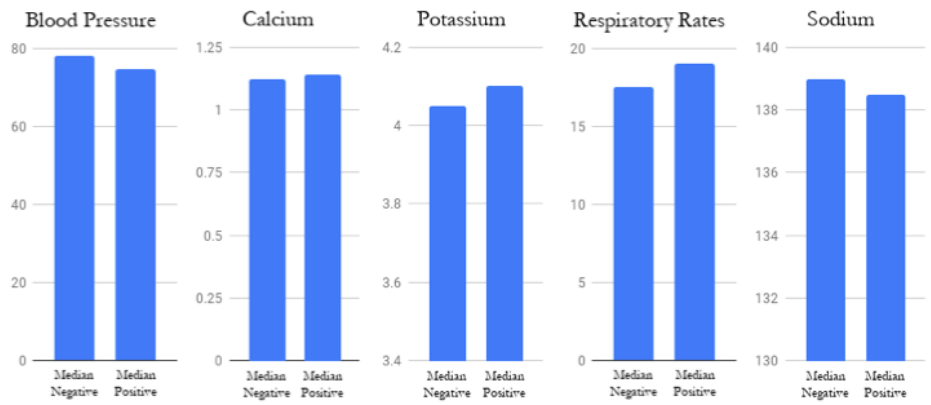

Figure 5.4: Comparison of median blood pressure, calcium, potassium, respiratory rate, and sodium values, where positive represents patients diagnosed with arrhythmia and negative represents those without [34].

shortly before the time of first medication, but it is possible that a record of high serum values could have been a result of the treatment rather than a cause of the diagnosis.

Additionally, many of the medications used for determining the time frame of data that should be used for a diagnosed patient were antiarrhythmiac medications that act as sodium-, calcium-, and potassium-channel blockers. These medications are typically administered when a patient's sodium, potassium, or calcium levels are too high. Since we measured only the hours immediately before drug administration, it could be that during these hours patients had an especially high amount of serum sodium, potassium, or calcium in their blood, further exaggerating their importances.

The model's accuracy is also highly dependent on the amount and quality of input data [23]. Specifically, the accuracy of the model is reliant on a reliable and consistent ground truth [23]. A pitfall of using the MIMIC-III database is that only hospital patients are included which does not facilitate the comparison of patients diagnosed with arrhythmias against completely healthy people. The resulting discrepancies are difficult to identify and measure.

Chapter 6

Conclusion and Future Work

In this thesis, we quantified with Random Forest the predictive power of various biosignals, medications, and preconditions for the diagnosis of cardiac arrhythmias. We achieved an Area Under the Receiver Operating Curve (AUC) score of 0.98, comparable to the performances achieved in several of the published state-of-the-art studies. We substantiated claims that each of sodium, calcium, potassium, respiratory rates and blood pressure can be used for the early diagnosis of cardiac arrhythmias. Our experimental results suggest that machine learning approaches such as this one could in the future could aid in the diagnosis of cardiac arrhythmias.

In a further study, we propose manipulating the threshold, which is explained in Section 3.2, to minimize Type 1 Error, or False Positives. The incorrect classification of a healthy patient is a particularly negative result, as the once healthy patient may become ill as a result of treatment for a disease they did not have. We would also recommend testing further features, such as magnesium.

Bibliography

[1] Heikki V. Huikiri, Agustin Castellanos, and Robert J. Myerburg. Sudden death due to cardiac arrhythmias. *The New England Journal of Medicine*, 345, 2001.

[2] Igor Splawski, Katherine W. Timothy, Michihiro Tateyama, Colleen E. Clancy, Alka Malhotra, Alan H. Beggs, Francesco P. Cappuccio, Giuseppe A. Sagnella, Robert S. Kass, and Mark T. Keating. Variant of SCN5A Sodium Channel Implicated in Risk of Cardiac Arrhythmia. *Science*, 2017.

[3] The Cleveland Clinic. Sudden Cardiac Death, 2019. Available at: `https://my.clevelandclinic.org/health/diseases/17522-sudden-cardiac-death-sudden-cardiac-arrest`.

[4] American College of Cardiology. Apple heart study identifies AFib in small group of Apple watch wearers, 2019. Available at: `https://www.acc.org/latest-in-cardiology/articles/2019/03/08/15/32/sat-9am-apple-heart-study-acc-2019`.

[5] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24, 2016.

[6] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. *Journal of Machine Learning Research*, 56, 2016.

[7] Marzyeh Ghassemi, Marco A.F. Pimentel, Tristan Naumann, Thomas Brennan, David A. Clifton, Peter Szolovits, and Mengling Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. *AAAI Conference on Artificial Intelligence*, 2015.

[8] Matthew M. Churpek, Richa Adhikari, and Dana P. Edelson. The value of vital sign trends for detecting clinical deterioration on the wards. *Elsevier*, 2016.

[9] Patrick Schwab, Gaetano C Scebba, Jia Zhang, Marco Delai, and Walter Karlen. Beat by Beat: Classifying Cardiac Arrhythmias with Recurrent Neural Networks. 2017.

[10] Marzyeh Ghassemi, Nicole Brimmer, Rohit Joshi Tristan Naumann, Peter Szolovits, Finale Doshi-Velez, and Anna Rumshisky. Unfolding physiological state: Mortality modelling in intensive care units. *KDD*, 2014.

[11] Carol Ann Remme and Connie R. Bezzina. Sodium Channel (Dys)function and Cardiac Arrhythmias. *Cardiovascular Therapeutics*, 28, 2010. doi: 10.1111/j.1755-5922.2010.00210.x.

[12] Jeremy Pinnell, Simon Turner, and Simon Howell. Cardiac muscle physiology. *Continuing Education in Anaesthesia Critical Care and Pain*, 7:85—88, 2007.

[13] Mark T. Keating and Michael C. Sanguinetti. Molecular and cellular mechanisms of cardiac arrhythmias. *Cell*, 104, 2001.

[14] Sana M. Al-Khatib, William G. Stevenson, Michael J. Ackerman, William J. Bryant, David J. Callans, Anne B. Curtis, Barbara J. Deal, Timm Dickfeld, Michael E. Field, Gregg C. Fonarow, Anne M. Gillis, Christopher B. Granger, Stephen C. Hammill, Mark A. Hlatky, Jose A. Jogla, G. Neal Kay, Daniel D. Matlock, Robert J. Myerburg, and Richard L. Page. 2017 AHA/ACC/HRS Guideline for Management of Patients with Ventricular Arrhythmias and the Prevention of Sudden Cardiac Death: Executive Summary. *Journal of the American College of Cardiology*, 72, 2018.

[15] Jose A. Jogla, Richard L. Page, Mary A. Caldwell, Hugh Calkins, Jamie B. Conti, Barbara J. Deal, N.A. Mark Estes III, Michael E. Field, Zachary D. Goldberger, Stephen C. Hammill, Julia H. Indik, Bruce D. Lindsay, Brian Olshansky, Andrea M. Russo, Win-Kuang Shen, Cynthia M. Tracy, and Sana M. Al-Khatib. 2015 AHA/ACC/HRS guideline for management of patients with supraventricular tachycardia. *Journal of the American College of Cardiology*, 2015. doi: 10.1161/CIR.0000000000000311.

[16] Sébastien Krul. Cardiac arrhythmias, 2013. Available at: `https://www.textbookofcardiology.org/wiki/Cardiac_Arrhythmias`.

[17] Harvey Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, and James Darnell. *Molecular Cell Biology*. W.H. Freeman, New York, New York, 2000.

[18] Michael C. Sanguinetti and Martin Tristani-Firouzi. hERG potassium channels and cardiac arrhythmia. *Nature*, 440, 2006.

[19] Igor Splawski, Katherine W. Timothy, Niels Decher, Pradeep Kumar, Frank B. Sachse, Alan H. Beggs, Michael C. Sanguinetti, and Mark T. Keating. Severe Arrhythmia Disorder Caused by Cardiac L-type Calcium Channel Mutations. *Proceedings of the National Academy of Sciences of the United States of America*, 102, 2005.

[20] John F.Fieselmann, Michael S. Hendryx, Charles M. Helms, and Douglas S. Wakefield. Respiratory Rate Predicts Cardiopulmonary Arrest for Internal Medicine Inpatients. *Journal of General Internal Medicine*, 8, 1992. doi: 10.1007/bf02600071.

[21] Michelle A Cretikos, Rinaldo Bellomo, Ken Hillman, Jack Chen, Simon Finfer, and Arthas Flabouris. Respiratory rate: the neglected vital sign. *The Medical Journal of Australia*, 188, 2002.

[22] Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. MIMIC-III, a freely accessible critical care database. 2016. doi: 10.1038/sdata.2016.35.

[23] Gurpreet Singh, Subhi J. Al'Aref, Marly Van Assen, Timothy Suyong Kim, Alexander van Rosendael, Kranthi K. Kolli, Aeshita Dwivedi, Gabriel Maliakal, Mohit Pandey, Jing Wang, Virginie Do, Manasa Gummalla, Carlo De Cecco, , and James K. Min. Machine learning in cardiac ct: Basic concepts and contemporary data. *Journal of Cardiovascular Computed Tomograph*, 2018. doi: 10.1016/j.jcct.2018.04.010.

[24] 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, and E. Duchesnay. Scikit-learn User Guide. pages 149–648, 2019. Available at: https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf.

[25] Microsoft Azure. How to choose algorithms for azure machine learning studio, 2019. Available at: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice.

[26] L. Breiman and A. Cutler. Random forests. *Machine Learning Journal*, 2002. Available at: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[27] Shagufta Tahsildar. Random Forest Algorithm in Trading Using Python, 2019. Available at: https://blog.quantinsti.com/random-forest-algorithm-in-python/.

[28] Will Koehrson. An Implementation and Explanation of the Random Forest in Python, 2018. Available at: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76.

[29] Seema Singh. Understanding the Bias-Variance Tradeoff, 2018. Available at: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229.

[30] Sarang Narkhede. Understanding AUC-ROC curve, 2018. Available at: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[31] Haiyan Huang. Multiple Hypothesis Testing and False Discovery Rate, 2018. Available at: https://www.stat.berkeley.edu/~hhuang/STAT141/Lecture-FDR.pdf.

[32] Center for Disease Control and Prevention. International classification of diseases, ninth revision (ICD-9), 2015. Available at: https://www.cdc.gov/nchs/icd/icd9.htm.

[33] SQLite release (3.27.2), 2019. Available at: https://www.sqlite.org/changes.html.

[34] T. J. Pollard and A. E. W. Johnson. The MIMIC-III Clinical Database. 2016. doi: http://dx.doi.org/10.13026/C2XW26.

[35] American Heart Association. Heath topics, 2017. Available at: https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings.

[36] 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] Python Software Foundation. Python language reference, version 2.7, 2018. Available at: https://docs.python.org/2.7/.

[38] SciPy Organization. Numpy reference guide, version 1.16, 2019. Available at: https://docs.scipy.org/doc/numpy/about.html.

[39] Google Development. Classification: ROC Curve and AUC, 2019. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[40] Edgar Lerma, Eric Simon, and Alina Sofronescu. Laboratory medicine, 2018. Available at: https://reference.medscape.com/guide/laboratory_medicine.