# Time-series Analysis of Medical Intensive Care Unit Data

Isabel Margolis, Ledri Thaqi, Elfat Esati

July 26, 2020

# Contents

# Chapter 1

# Introduction

The intensive care units (ICUs) in hospitals suffer from a large number of false alarms. In the ICUs, there are continuous measurements of patient parameters such as heart rate, respiratory rate, and blood pressure. Currently, the system raises an alarm whenever a signal reaches a defined maximum or minimum threshold. Unfortunately, this results in many false alarms. A high number of false alarms is a big problem because it can lead to alarm fatigue. Hence, hospital staff becomes desensitized to alarms. Additionally, too many alarms can cause noise disturbances, which can lead to sleep deprivation for the patients [14].

In this project, we explored data from patients in the ICU, especially signals measured over time, such as heart rate, respiration, and pulse. Our goal was to observe how those signals change over time, correlate with each other, and to discover what could be a significant indicator for raising alarms. Secondly, our goal was to explore three different models and discuss whether these models could be beneficial in reducing the number of false alarms.

The first model involves the algorithm called ARIMA, short for 'Autoregressive Integrated Moving Average.' We introduce this model in chapter 4, which will provide an extensive discussion on whether or not ARIMA could be useful for decreasing false alarms in ICUs. The second model concentrates on correlations of multiple signals and how these correlations change over time. This model will be introduced in chapter 5. The third model focuses on the cross-correlation of different windows within a signal, which we present in chapter 6. For each model, we tested multiple methods and compared them with each other to investigate how and if these models could be useful in an ICU.

# Chapter 2

# Exploratory Analysis

We explored several different databases in this project. Each database had some advantages over the others, and we were able to gain valuable knowledge by spending much time on finding and exploring different databases. In general, this process helped us formulate the goal of this project and its structure.

We prepared our preliminary work with the help of the MIMIC-III Database [24]. This work included getting familiar with the medical data and exploring various signals and their correlations. All the analysis from this database is in Appendix A.

Our primary focus, however, was on the database from the PhysioNet Computing in Cardiology Challenge 2015 [21]. In contrast to the MIMIC-III Database, the PhysioNet Challenge 2015 Database contains well-labeled alarms, which is necessary to build a predictive model.

Section 2.1 will give an introduction and overview of the database, and section 2.2 will provide an exploratory analysis that we conducted on the database before and during the creation of the predictive models.

## 2.1 PhysioNet Challenge 2015 Database

The PhysioNet Challenge 2015 Database is publicly available, along with the challenge results and several papers by participants explaining their approaches [17].

The goal of the challenge was to create an algorithm to reduce high false arrhythmia alarm rates. The alarms are grouped into five different life-threatening arrhythmia events:

- Asystole (ASY): There are no heartbeats for a period of four seconds or more.
- Extreme Bradycardia (EBR): The patient's heart rate is lower than 40 beats per minute; fewer than five beats occur within a period of six seconds.
- Extreme Tachycardia (ETC): The heart rate is higher than 140 beats per minute; more than 17 beats occur within a period of 6.85 seconds.

- Ventricular Tachycardia (VTA): There are five or more consecutive ventricular beats within a period of 2.4 seconds (a rate of 100 per minute.)
- Ventricular Fibrillation or Flutter (VFB): The heart exhibits a rapid fibrillatory, flutter, or oscillatory waveform for at least four seconds.

The database consists of 750 records, and each record contains up to 4 signals. There are always two ECG Leads present (I, II, III, or V) and at least one of the following signals: Arterial Blood Pressure (ABP), Pulse (PLETH) and Respiration (RESP). Each data signal is 5 minutes long, ending at the time of the alarm. At least two experts labeled the alarms. The label is 0 for a false alarm, and 1 for a true alarm.

| | II | V | PLETH | RESP |
|---|---|---|---|---|
| 0 | 0.453 | 0.381 | 0.436 | -0.188 |
| 1 | 0.501 | 0.417 | 0.469 | -0.205 |
| 2 | 0.467 | 0.381 | 0.420 | -0.187 |
| 3 | 0.505 | 0.408 | 0.443 | -0.200 |
| 4 | 0.523 | 0.418 | 0.447 | -0.204 |

Table 2.1: Example of Record a186s

Table 2.1 shows an example of the first five rows for the patient in record a186s. To illustrate the following methods and analyses, we will use the patient in record a186s. That patient had a false asystole alarm.

Each record contains 75'000 rows, and each signal was sampled at a frequency of 0.004 seconds.

The participant's task of the PhysioNet Challenge 2015 was to determine which alarms represented true arrhythmias, and which were caused by other factors (such as noise, patient movement, leads falling off, or misidentification of ECG features on the part of the monitor) [17].

The scores of the participants are published. Unfortunately, the hidden test set for the participants is not available. Therefore, we were unable to replicate their results. The scores are computed using the following formula:

$$score = \frac{100 * (TP + TN)}{TP + TN + FP + 5 * FN} \tag{2.1}$$

False negatives are weighted five times more harshly because genuinely life-threatening events classified as false alarms can be hazardous for the patient.

## 2.2 Exploratory Analysis

Table 2.2 shows the number of records that contain the corresponding signal. Since III, MCL, I, aVF, aVR, and aVL do not have a large count, we did not consider those signals for most of the analysis.

| Signal | Count |
|---|---|
| II | 728 |
| V | 684 |
| PLETH | 627 |
| ABP | 343 |
| RESP | 278 |
| III | 39 |
| MCL | 28 |
| I | 13 |
| avF | 3 |
| aVR | 3 |
| aVL | 2 |

Table 2.2: Number of Records per Signal

| | **Alarm Types** | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| True Alarms | 22 | 46 | 131 | 6 | 252 |
| False Alarms | 100 | 43 | 9 | 52 | 89 |
| Total | 122 | 89 | 140 | 58 | 341 |

Table 2.3: Number of Records per Alarm Type

Table 2.3 gives an overview of the number of records in the database for each alarm type.

Figure 2.1 shows a side-by-side example of a record with a true asystole alarm and one with a false asystole alarm. There are apparent differences in the signals. However, most records in the database have very different signals. In general, records of false alarms seem noisier, and the y-axis, i.e., the value of the vital sign, often has a wider range for false alarms than true alarms, as can be seen in Figure 2.1.

### 2.2.1 Exploratory Analysis from the Correlation Perspective

In this section we unveil our results coming from the exploratory analysis based on the correlation of signals. Section 2.2.2 will provide an overview of the types of signals that are present in the PhysioNet Challenge 2015 Database. In Section 2.2.3 we will explore the signal correlations for each alarm type.

### 2.2.2 Different Types of Signals and their Meanings

Every record consists of up to four signals, including two ECG leads and up to two other signals, including ABP, PPG, or respiration. The two ECG signals present

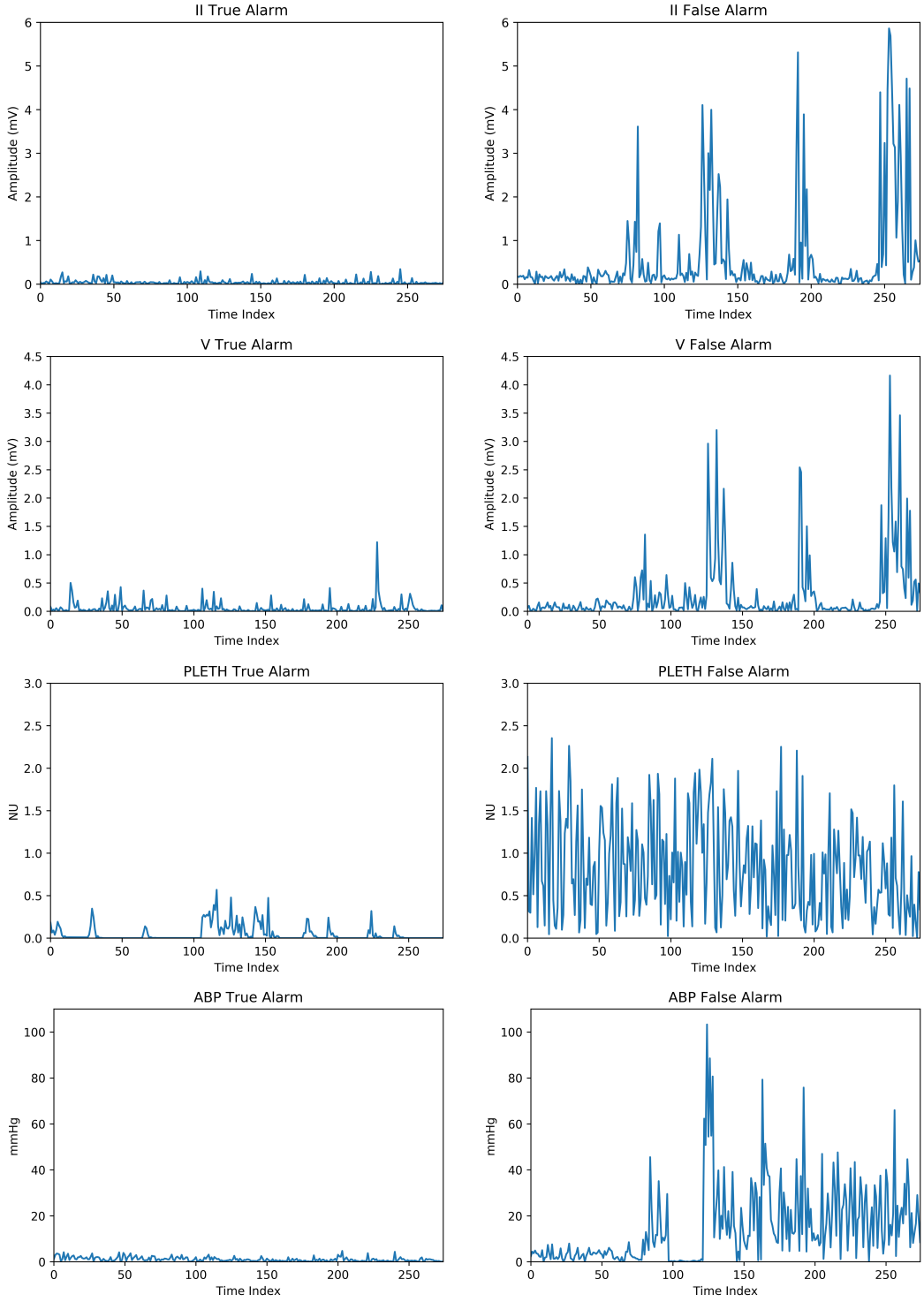



Figure 2.1: Record Examples of True and False Alarms

in the PhysioNet Challenge 2015 Database include ECG II and ECG V, which are among the 12 ECG leads.

The 12 ECG leads draw a complete picture of the heart's electrical activity captured from 12 different perspectives [1]. The 12 leads constitute the 12 different angles of which the heart can be viewed, making a cohesive picture of the heart.

The 12 ECG leads capture the electrical activity of the heart measured in amplitude. In Figure 2.2, we can see an example of signals showing the normal flow of signals of a healthy patient. The signals shown are also present in the PhysioNet Challenge 2015 Database. In ECG, each square represents a time interval of 0.04 seconds [2]. All variants of ECG signals represent the heart rhythms, but other signals only indirectly represent the patient's condition. Since all alarms in this dataset are raised due to arrhythmia events or irregular heartbeats, the main signals important to us are the ECG leads.


Figure 2.2: Example of normal tracing of signals that are also present in the PhysioNet Challenge 2015 Database. *Source: [25]*

The ECG leads depend on the position of electrodes attached to the chest, wrist, or arm [35]. A lead which is composed of two electrodes of opposite polarity is called a bipolar lead. On the other hand, a lead consisting of a single positive electrode is a unipolar lead. ECG II is a bipolar lead, while ECG V is a unipolar lead. Each ECG signal, independent of which lead or perspective is applied, consists of three main components which are also shown in Figure 2.3:

- P-wave
- QRS
- T-wave

No matter which ECG lead is applied, the QRS wave is always present, assuming the patient is healthy. That also means that in ECG V, we can typically see the QRS wave [22]. The components of ECG are the core elements that show a stable and healthy heart. The P-wave represents the depolarization of the atria. The

QRS represents ventricular depolarization, and the T-wave represents repolarization [19]. After we explore the events that caused each alarm, we will know which components are changing and what type of alarm is raised.

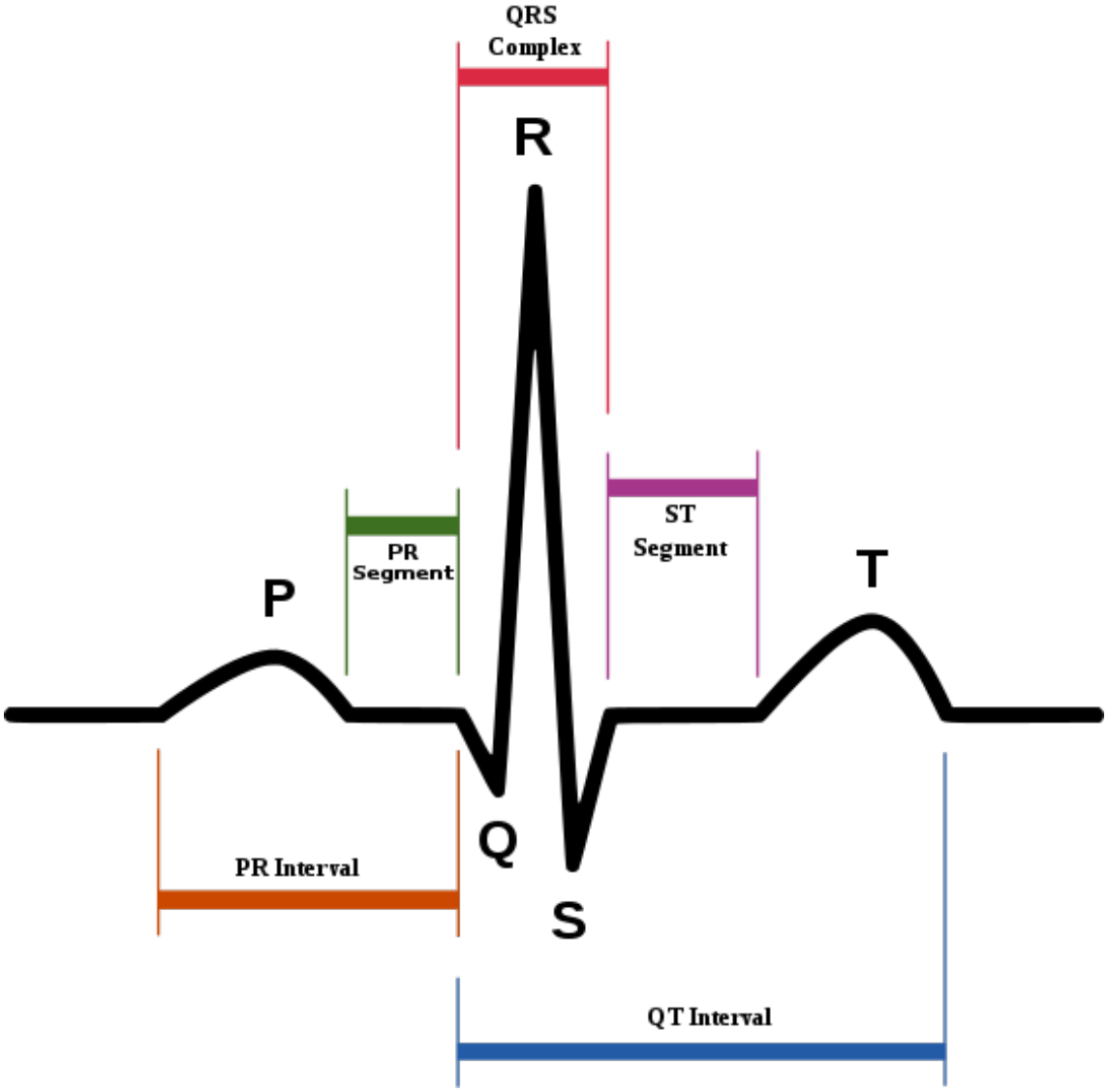



Figure 2.3: Normal sinus rhythm of heart. *Source: [7]*

### 2.2.3 Exploring Records Based on Alarm Type

The basic idea is to explore records based on the type of alarm being raised. First of all, a basic correlation has been applied to determine how strongly signals are correlated with each other. By randomly picking up records, we found that ECG II and ECG V are highly correlated either positively or negatively. This happens because in the ECG V, either the positive or negative polarity has been attached to the body.

After filtering out the records with one alarm type, we then compared the records with true alarm and false alarm.

### Asystole Alarms

In this section, we explore the records with asystole alarms. Asystole is one of the most severe conditions which occurs due to the heart ceasing to beat. In the case of the PhysioNet Challenge 2015, an alarm is issued when that condition exceeds 4 seconds. All records with asystole alarms were filtered and explored. The alarms are raised at the end of the record. Each record has 75000 rows or 5 minutes of real-time. The event that triggers the alarm occurs in the last 10 seconds of a record, which means that it must happen in the last 2500 rows. Consequently, we extracted the last 10 seconds and plotted them out.

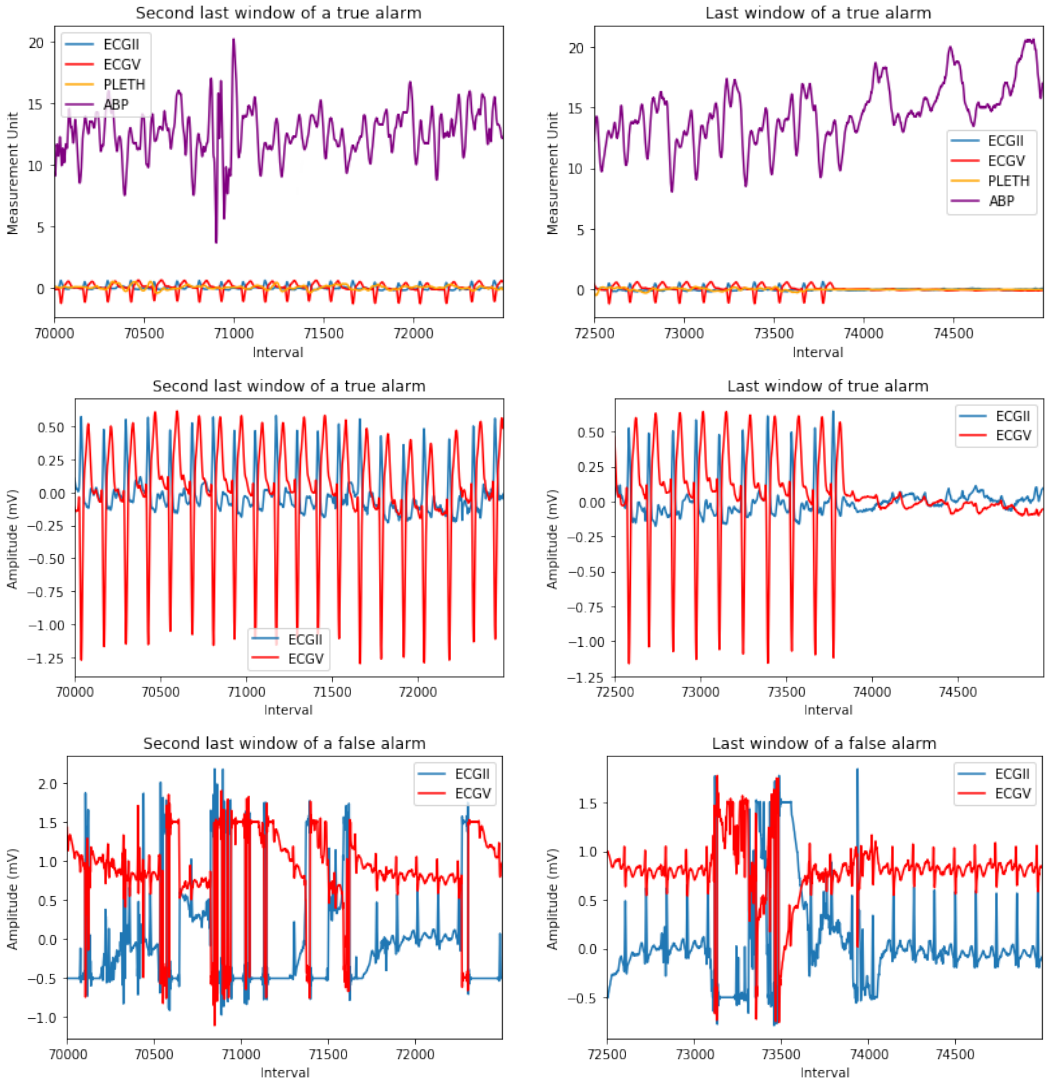



Figure 2.4: Assystole alarms

Figure 2.4 shows a true alarm and a false alarm patient extracted from two different records. On the left-hand side is the second last window of a record or the window that begins at 04:40 until 04:50, and on the right-hand side is the last window which begins at 04:50 and ends at 05:00. In the first row, we see all the signals present in a patient's record with a true alarm. When comparing the windows, we

see that in the right window, the oscillation changes and the ECG II and ECG V signals become flat. This shows that there are no more QRS and no more peaks, which means the heart has stopped functioning and eventually an asystole alarm has taken place.

The second row of Figure 2.4 shows the same patient but now only the ECG II and ECG V signals. The figure shows that the last window no longer has peaks. In contrast, the last row of the figure is of a patient with a false asystole alarm. Although the signals seem to be distorted in both windows, a heartbeat still exists. The heart still beats, although the beats are not regular and in lower amplitude. Therefore no asystole alarm should have been issued.

The ECG signals in the false alarm patient are negatively correlated, unlike the patient with true alarm. As we have mentioned earlier, this is due to the change of position of the electrode attached to the body of a patient.

### Extreme Bradycardia Alarms

Bradycardia alarms are raised if the heart rate is lower than 40 bpm or less than five beats occur within six seconds. Depending on the patient's age, the low heart rate can be qualified differently. In Figure 2.5, similar as in the previous figure, we have chosen a true alarm and a false alarm patient. In the first two figures, we have the second last and last window, on the left and the right, respectively, of a patient with true alarm with all the signals. Below those figures, we can see more clearly the same windows but only with the ECG signals. In the last row, we have a patient with a false alarm. From the figure, we can see that the number of QRS waves appear more frequently than in the true alarm window. Therefore this confirms that a Bradycardia alarm should not have been raised, although in the last window, the waves are not consistent. However, that does not mean that there are fewer heartbeats, but rather the signals do not form a complete QRS wave.

### Extreme Tachycardia Alarms

In this type of alarm, the number of heartbeats is higher than 140 beats per minute, or in other words, an alarm is raised if the heart rate exceeds 17 beats within a period of 6.85 seconds. In Figure 2.6 we can see another extract of a record with true and false Tachycardia alarm. Different signals appear in different records. However, the ECG signals appear 95 percent of the time. Due to that, we have plotted the true alarm case with all signals and with ECG signals only. Since the ECG signals appear in all records, the ECG signals can be seen in a true record and a false alarm record for comparison purposes.

In the true alarm record, the number of heartbeats appears to be more frequent than the number of heartbeats in the false alarm record. Although the ECG signals differ in amplitude, the number of heartbeats appear simultaneously in a record. Both ECG signals measure the QRS complex of the heart, but at different angles. Therefore there is a high correlation between the ECG II and ECG V signals. In the false alarm record, the signals have different amplitudes, but that does not necessarily mean that the signals do not correlate.

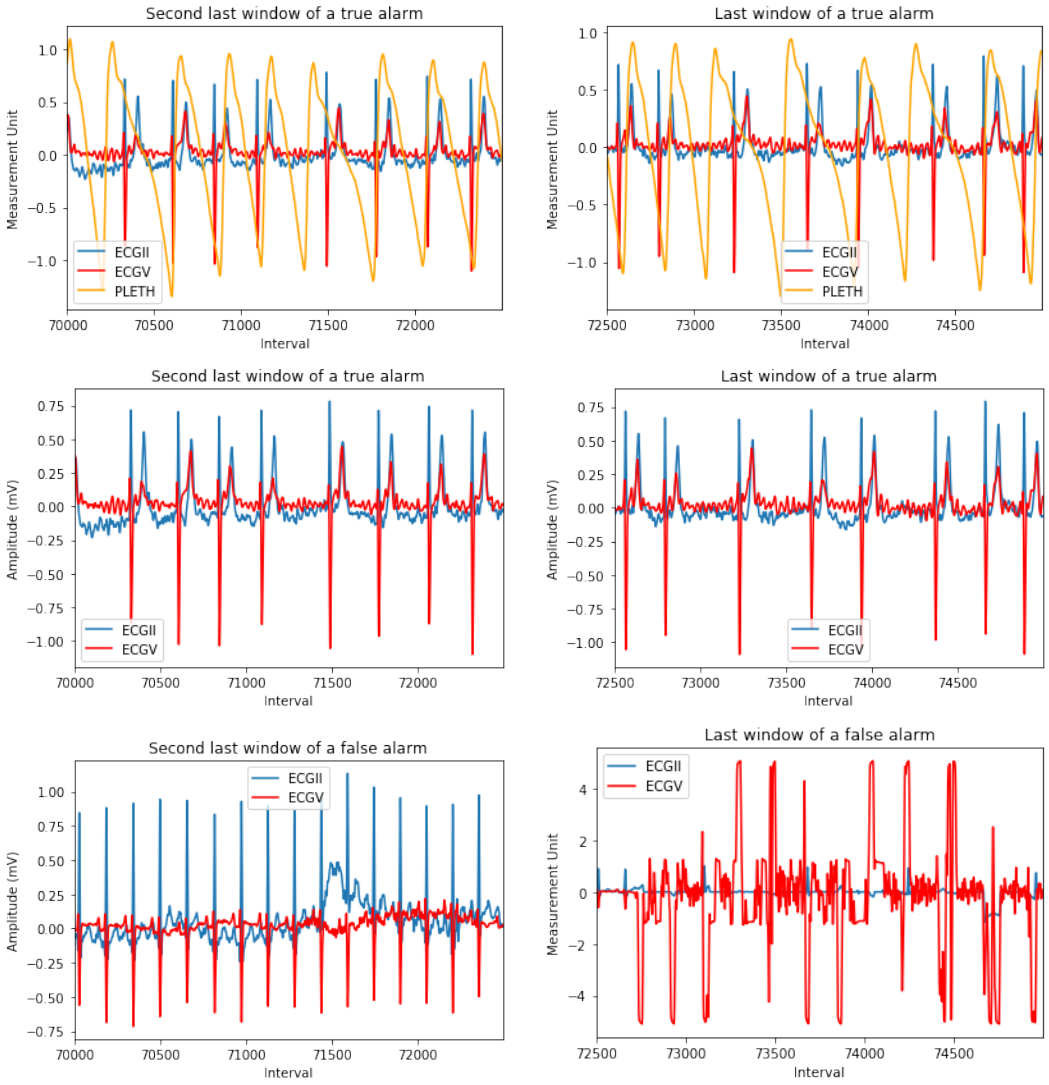



Figure 2.5: Bradycardia alarms

### Ventricular Tachycardia Alarms

Ventricular Tachycardia alarms are raised when there are five ventricular beats with a heart rate higher than 100 bpm [10]. Similarly, this alarm type is described as having five or more consecutive ventricular beats within a period of 2.4 seconds. Here ventricular beat does not mean regular heartbeats but heartbeats of a different shape of waves. We have mentioned earlier the different components that form a proper heartbeat. In this type of alarm the p-wave is usually not present, although the rhythm may be regular. Another wave that changes is the QRS wave, which becomes wider or takes more time than a normal QRS wave. In Figure 2.7, we can see that the true alarm happens in the last window of 10 seconds with a more sinusoidal wave that does not show a heart that rests and beats again. Similarly, the correlation between the ECG signals also follows. In the false alarm record there is no sinusoidal like wave and all the components of a complete ECG signal are present although the number of beats seem to be a bit higher.

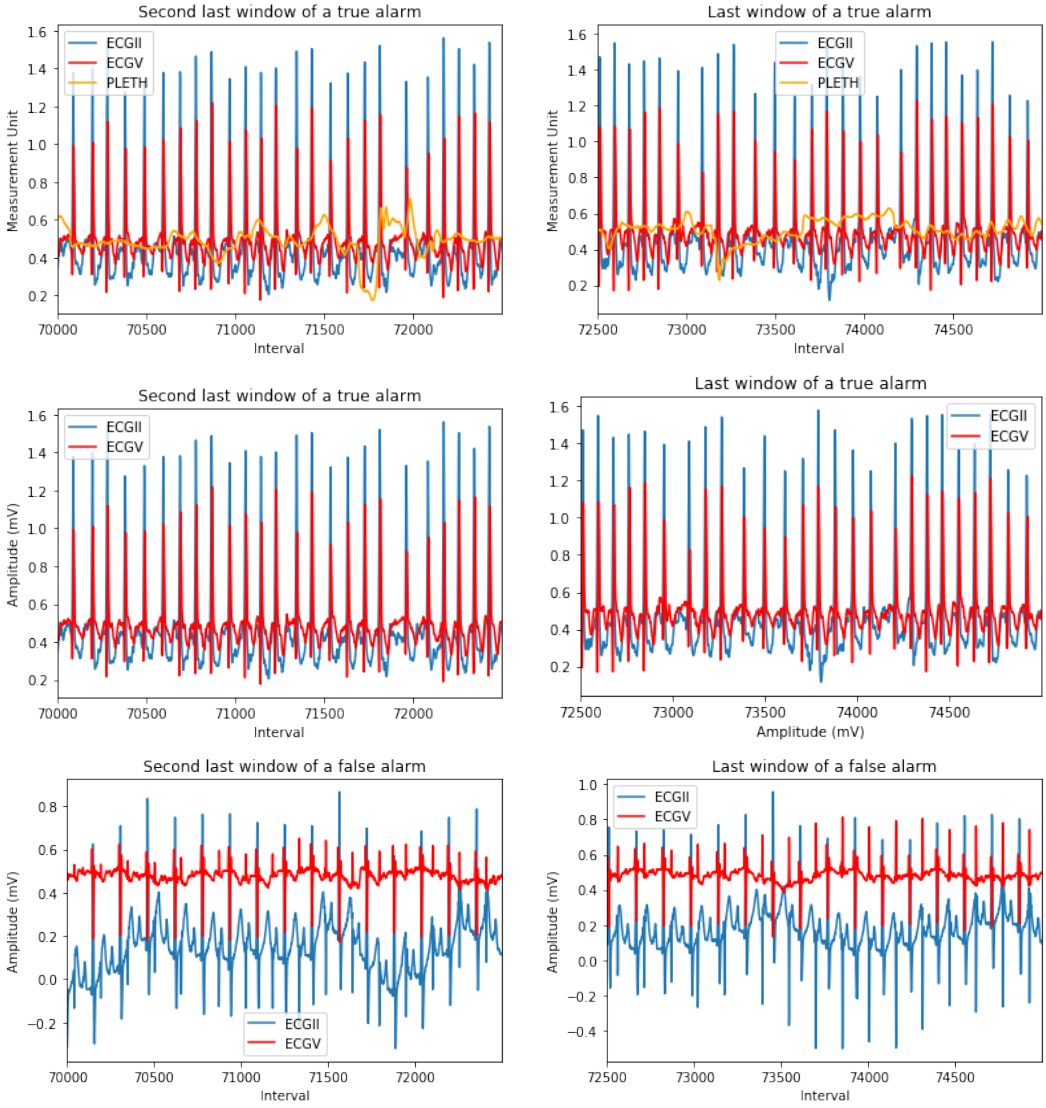



Figure 2.6: Tachycardia alarms

### Ventricular Fibrillation or Flutter

This type of alarm detects the chaotic electrical activity in ventricles, which results in the quivering of ventricles. There is no regular rhythm, p-waves and QRS are missing. This alarm is best described in Figure 2.8 of a record with a true alarm where the waves are distorted and do not form a steady rhythm. The amplitude of the signals has also changed, and there is little to no correlation between the signals. In contrast, in the false alarm record, the ECG waves are still present, but the heart rate has increased. Even though there seems to be a faster heart rate, the alarm does not belong to this category of alarms.

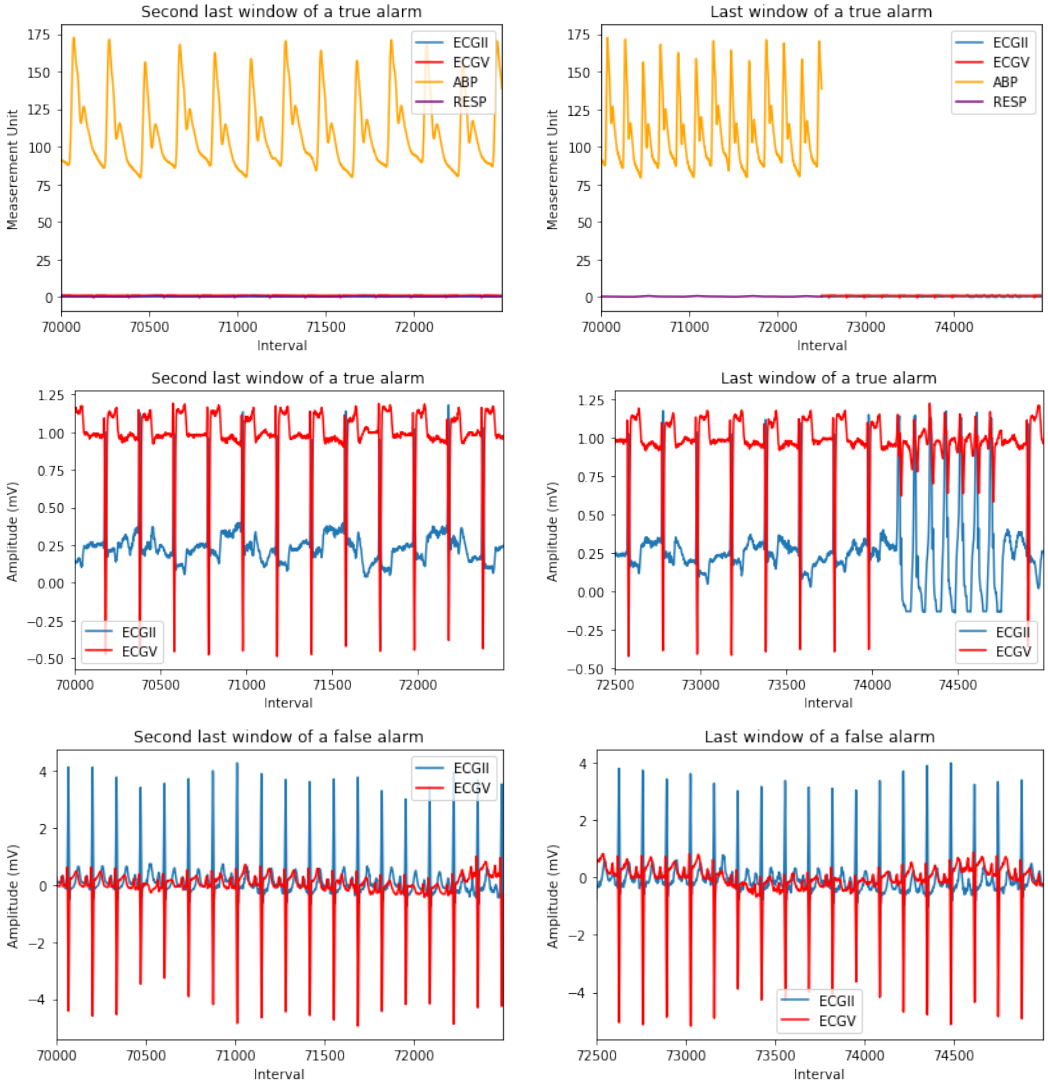



Figure 2.7: Ventricular Tachycardia alarms

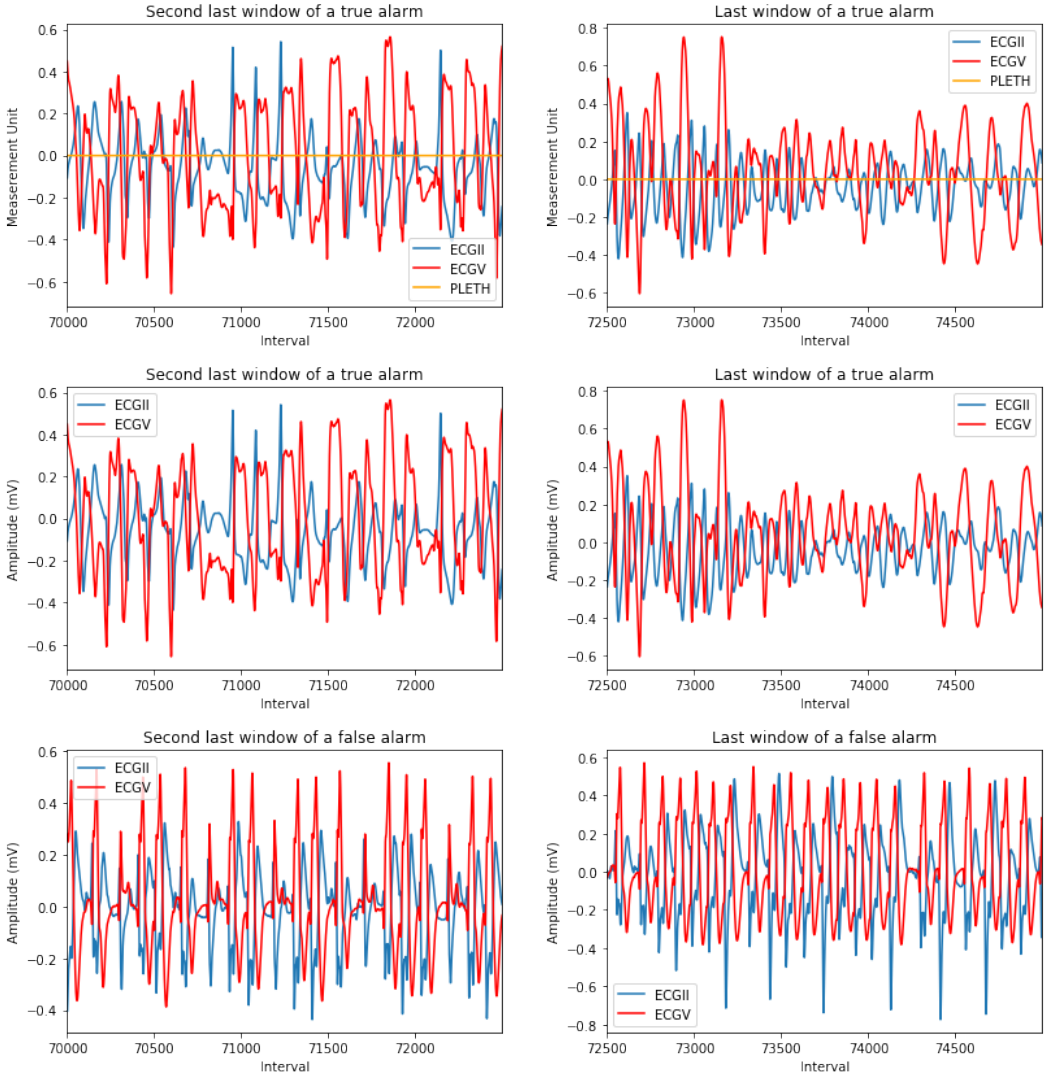



Figure 2.8: Ventricular fibrillation or flutter alarms.

### 2.2.4 Exploratory Analysis from the Window Correlation Perspective

Our approach to the exploratory analysis of data from the window correlation perspective was to treat the patient's signal as a large number of non-over-lapping windows and targeting interesting windows to cross-correlate them. The reason behind this approach was to examine if the cross-correlation of different windows within the same signal would yield any signs or patterns before the alarm happened. Thus, different signals at different window sizes were examined.

Additionally, as a base for the window approach and the determination of window sizes, we tested several time shifts to the signal in order to check for metrics such as seasonality, trend, and correlation. Using the seasonal decompose, these metrics were thoroughly observed and explored as they are the most common data variations that are present in a time series.

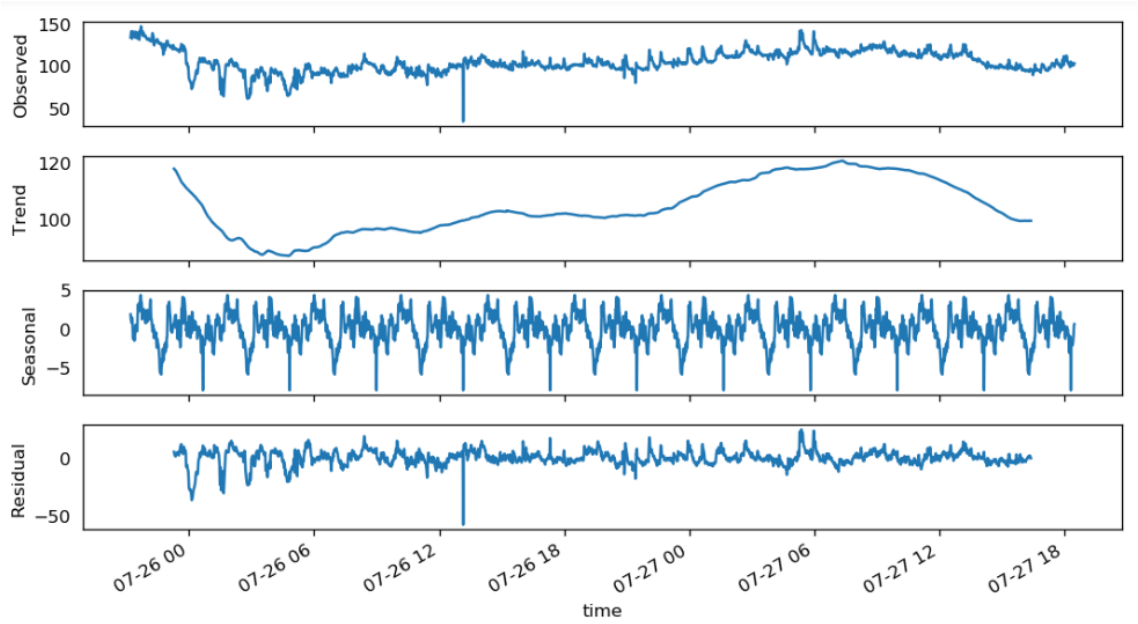



Figure 2.9: Decomposition analysis of ECG V signal

The ECG V lead signal which is represented by the "observed" plot of the figure is decomposed into trend, seasonality and residuals components which can be seen under their respective plots in Figure 2.9. Thus, these components can be added together to reconstruct the data shown in the top plot.

The trend component shows the overall movements in the signal while ignoring the seasonality and small random fluctuations such as the one in the observed top plot. Ideally the trend of a signal should capture most of the patterns in the signal since it is focused only on the variations of low frequency in a signal. Moreover, it visualizes the general tendency of the data to increase or decrease during a long period of time, in this case starting when the signal is being recorded up to the alarm being raised. In our signals, it can be observed from the pattern of the trend that our trend is non linear since it does not follow a straight line. This non linear trend suggests that the values within the signal which later will be represented

by windows follow a downward, stable and upward pattern throughout the whole signal.

The seasonality component presents the cycles and regular fluctuations that are repeated over time. These cycles are considered seasonal only if they only repeat at the same frequency, direction and magnitude, as also seen on the seasonality plot on Figure 2.9. The seasonality can be either removed resulting in a clearer relationship between the input and output variables, or left untouched as it was in our case to gather more information about the signal and improve our model performance.

Lastly, taking into consideration that trend and seasonality capture most of the patterns in a signal, if we remove trend and seasonality from the signal then what is left are the residuals. Residuals usually account for patterns that move too fast and do not obey the timing of the signal. Thus, the pattern cannot be assigned either to trend (pattern moves too fast for the trend component) or seasonality (pattern does not obey timing of seasonal component). The residual plot on Figure 2.9 shows exactly those patterns and an example of an outlier which may potentially be due to sensor failure, noise or other errors.

Subsequently, examining the above metrics proved to e helpful when examining the correlation since we were able to track and identify correlation patterns across the signal. We correlated the signal with itself by shifting the data points by t+1, t+2, t+3, and t+4. Respectively, the data points are correlated with their time **t** shifted data points. We observed that as we increase the lag (time shifts), the correlation decreases. On the other hand, as the lag decreases, the data points become closer to each other, as shown in Figure 2.10.

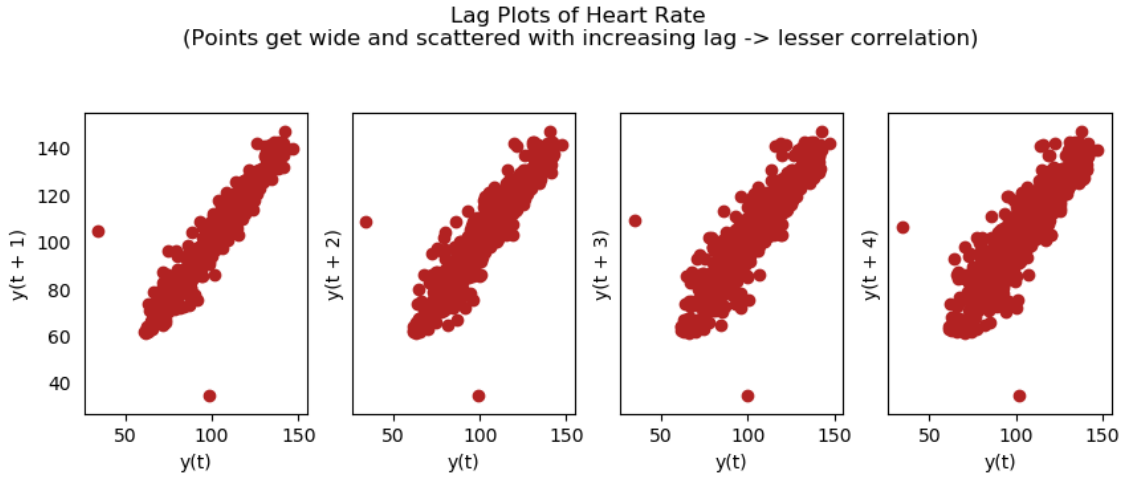



Figure 2.10: Correlation changing according to the lags

Furthermore, Figure 2.10 also shows that we can identify patterns of correlation between data points that are closer and weaker correlation between data points that are further between each other. These insights were taken into consideration later during the implementation of the model and tuning of the hyper parameters.

In Figure 2.11 we can see the plot of two windows of the same signal with their respective metrics such as respiratory, pleth and two ECG leads II and V. As we can

also observe from the plot the respiratory and the ECG lead V are highly correlated however ECG lead V experiences a sudden drop which might prove to be noise-related. Moreover, the correlation coefficient of 0.82 proves that these particular windows of the same signal are correlated, worth to be further explored and used as prime examples in our model.

On the other hand, in Figure 2.12 we observe two different windows of the same signal which are slightly correlated. Apart from focusing on the lines of the above-mentioned metrics, the correlation coefficient of .40 is a clear sign that these two windows are not highly correlated as the other pair.

The size of these windows in terms of values that they represent was initially planned to be of size 50, however on further testing and checking for correlation between windows it was decided to move to a larger size of the windows resulting in the second version of the model. Subsequently, the window size was increased to 100 as we assumed that a larger window size would capture more information about the signal that each window holds. Thus, each window that was cross-correlated with another window of the same signal in the back end of the cross-correlation function was using the respective 100 values of each window of the signal in the respective medical metrics.

These findings were highly considered in the structure and implementation of the cross-correlation of windows model, this approach was kept in mind as we did not want to lose any information through the values inside the windows and as such the window size was carefully decided to be 100 subsequently covering and including valuable information through the signal. Concluding, the exploration, visualization and correlation check from this period was decisive in the steps towards implementing and using the model to predict and reduce the false alarm rate.

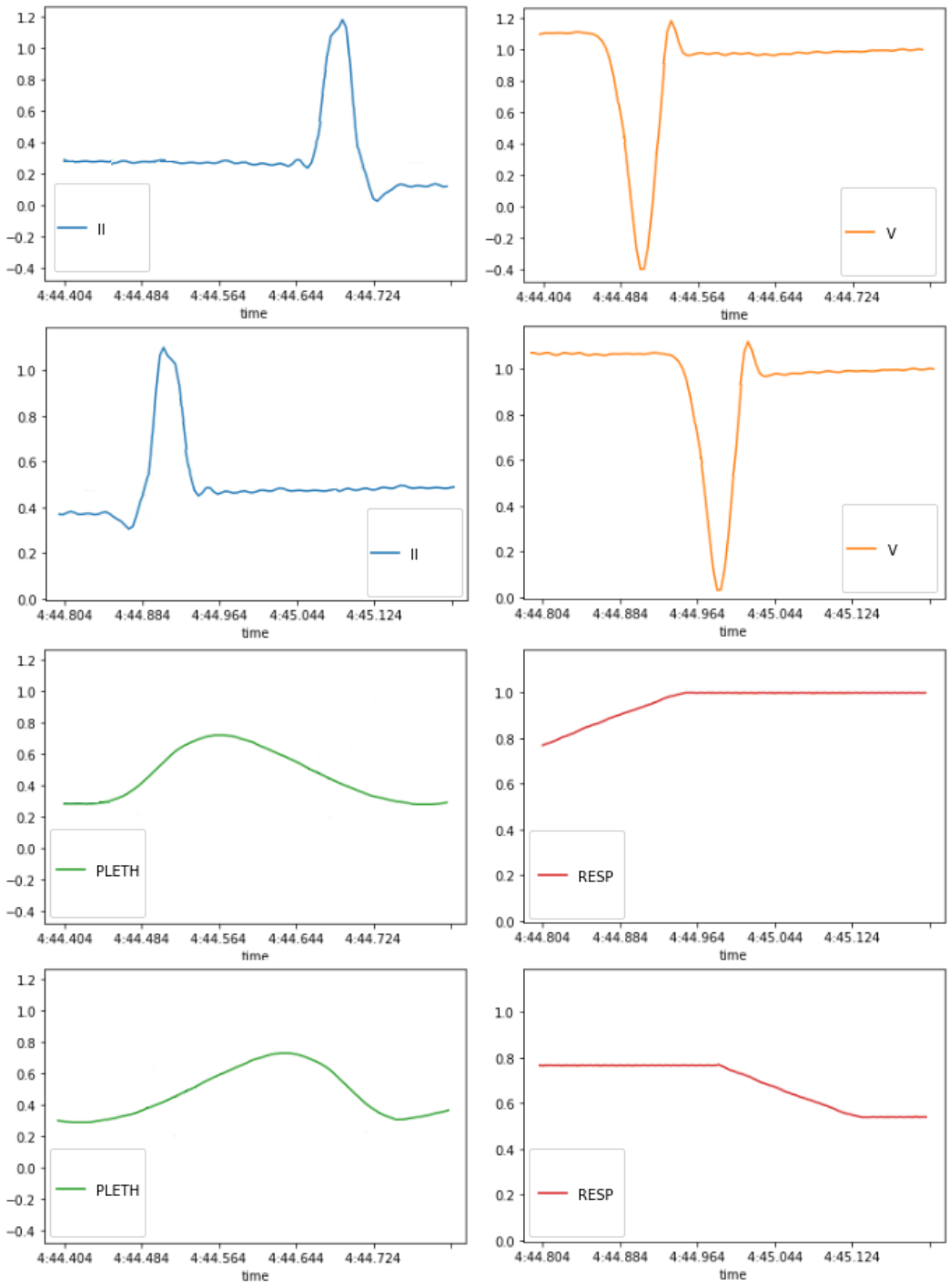



Figure 2.11: Highly correlated windows of the same signal

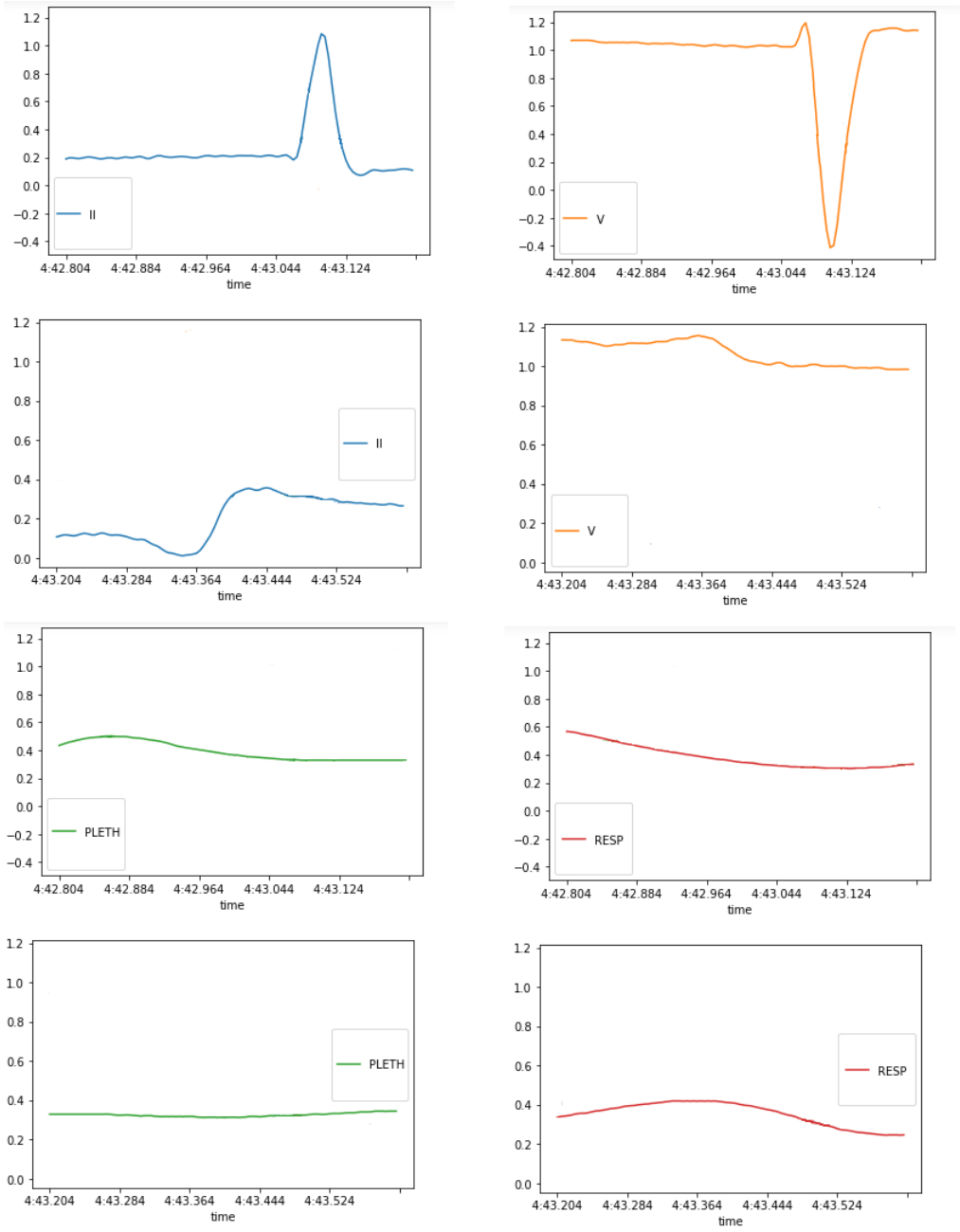



Figure 2.12: Slightly correlated windows of the same signal

# Chapter 3

# Signal Preprocessing

This chapter introduces a few signal preprocessing methods that were used by the competing teams in the PhysioNet Challenge 2015. Signals can suffer from bad signal quality if there is a lot of noise from movement artifacts, sensor disconnection, pacemaker or other noise artifacts. Low signal quality can cause false alarms. Therefore, it is important to address common issues that can arise and to spend a lot of time preprocessing the data. Most participants of the PhysioNet Challenge 2015 used many different preprocessing methods for the ECG leads in order to correctly identify peaks in the signals which corresponds to the heart rate. Those techniques include various QRS peak detection and extraction algorithms. The different methods proposed in this project do not deal with identifying peaks and therefore only the preprocessing techniques that deal with signal quality issues in general and not specific to QRS peak detections were considered. Additionally, only techniques that were used by the top ten scoring teams were examined.

## 3.1 Filtering Techniques

Filtering is one example of digital signal processing and its main goal is to reduce noise. Many different filtering techniques exist. The techniques that were most frequently used by the participants in the PhysioNet Challenge 2015 are introduced.

### 3.1.1 Low Pass Butterworth Filter

Low pass filters, in general, pass signals with a frequency lower than a certain threshold, called the cut-off frequency, gradually reduce the amplitude of the signals with a higher frequency. Short-term fluctuations have a high frequency and are therefore reduced to a lower frequency making the overall signal smoother than before. The Butterworth Lowpass Filter is popular, because it has a smooth response at all the frequencies and the signals are decreased monotonically from the cut-off frequency [4].

Figure 3.1 shows how the Butterworth Filter gradually decreases the signals with a higher frequency than the cut-off frequency. In the figure, the cut-off frequency is signified by the green vertical line at a frequency of $10^2$ Hz. Hence, all

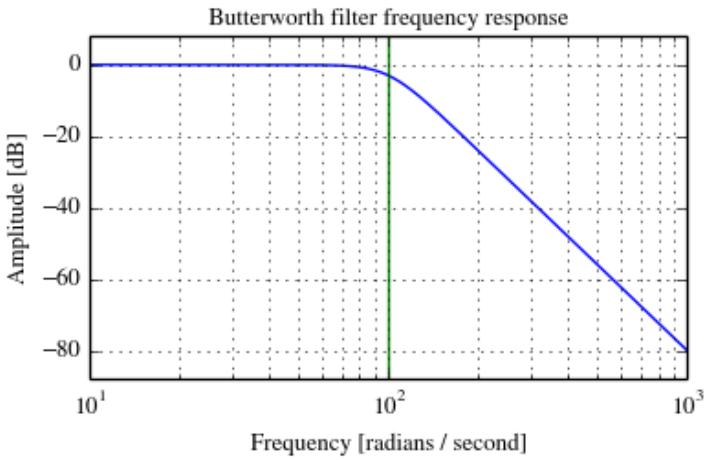



Figure 3.1: Butterworth Filter Frequency Response. *Source: [8]*

frequencies larger than $10^2$, in other words, all signals with more than 1000 cycles per second, are smoothed out to lower frequencies according to the monotonic curve seen in Figure 3.1.

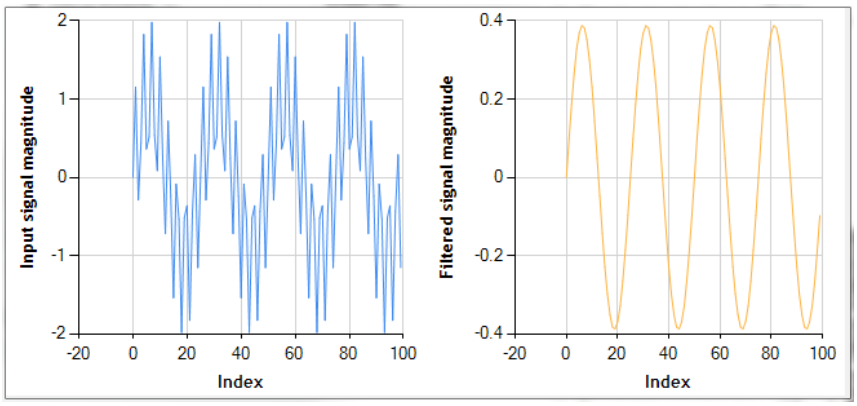



Figure 3.2: Butterworth Filter. *Source: [32]*

Figure 3.2 shows an example of a signal after using a Butterworth Lowpass Filter. The small noise from the higher frequency signal is gone and the signal with lower frequency remains.

### 3.1.2 Median Filter

The main idea of the median filter is to slide a window over the signal and replace the entry in the center of the window with the median of all the signals in that window [12].

Figure 3.3 shows an example of a median filter on a signal. The periodic shape of the signal remains while most of the noise is removed.

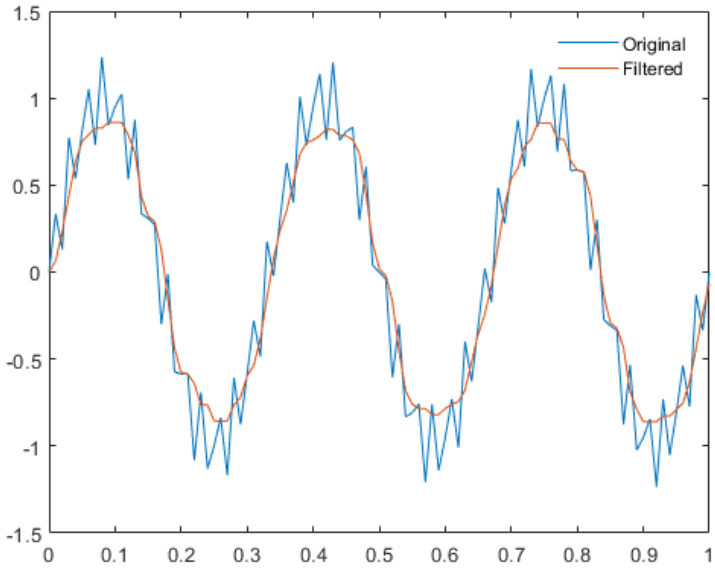



Figure 3.3: Median Filter. *Source: [26]*

### 3.1.3 Downsampling

Downsampling is the process of reducing the sample rate by an integer factor. A signal s[n] can be downsampled by a factor of Q by retaining every Qth sample and discarding the remaining samples. The new slower sample rate is 1/Q of the original faster sample rate [13]. [18] downsampled all signals to 125 Hz. This means that they used $Q = 2$ and discarded every second sample. Many other teams in the PhysioNet Challenge 2015 used $Q = 2$ as well.

## 3.2 Common Problems

The reports of the participants from the PhysioNet Challenge 2015 highlight two common problems that arise with ECG signals. The participants provide solutions to these problems, which were taken over in this project.

### 3.2.1 Baseline Wander

Baseline Wander can occur in ECG signals where the x-axis viewed on the screen appears to move up and down due to improper electrodes. This can cause an entire shift of the signal. [36] propose a method to remove Baseline Wander by applying a Butterworth Lowpass Filter to the ECG signal. By subtracting the filtered signal from the original signal it is possible to obtain a signal with almost zero Baseline Wander.

Figure 3.4 shows an example of Baseline Wander in the ECG signal for a record v136s in the PhysioNet Challenge 2015 Database. The image on the left shows the

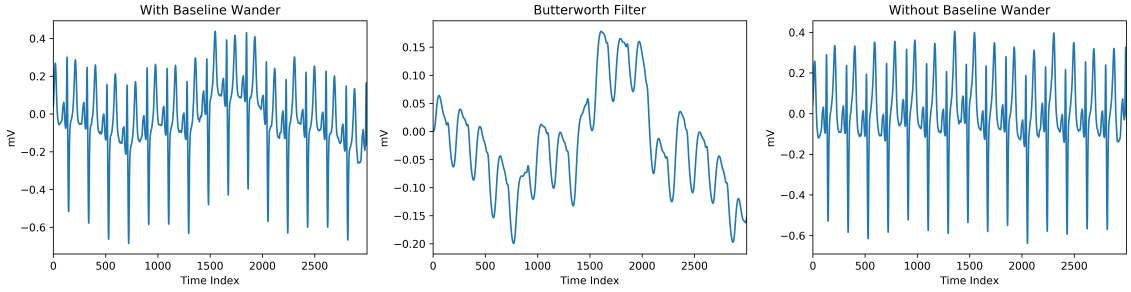



Figure 3.4: ECG II with Baseline Wander

original signal for a specific time interval. The image in the middle shows the signal after applying a Butterworth Lowpass Filter with a cut-off frequency of 1 Hz. This cut-off frequency was proposed by [36]. The image on the right shows the processed signal after subtracting the filtered signal from the original signal.

### 3.2.2 Pacemaker Spikes

Pacemaker spikes are vertical signals that represent the electrical activity of the pacemaker. Figure 3.5 shows an example of these spikes. To solve this issue [18] propose to compare each sample of an ECG signal with the average of the preceding and the following sample. If there is likely to be a pacemaker spike, they suggest to replace the corresponding samples through a linear signal interpolation.


Figure 3.5: Example of Pacemaker Spikes. *Source: [28]*

# Chapter 4

# ARIMA Models

Autoregressive Integrated Moving Average Models (ARIMA) can be used to model and forecast time-series. The goal of this chapter is to analyze whether an ARIMA model could be beneficial in reducing false alarms in the intensive care unit. The underlying idea is that ARIMA would be able to model a signal, i.e. heart rate, pulse, SpO2,... at times when the patient is stable, but as soon as the patient is in a critical state, ARIMA would produce higher errors and an alarm should be raised. In Section 4.1 we will provide a few important definitions and in Sections 4.3 and 4.4 we will introduce all the models based on ARIMA and discuss their results.

## 4.1 Definition

ARIMA(p,d,q) is a composition of the Auto Regressive Model (AR(p)) and the Moving Average Model (MA(q)). It can model any 'non-seasonal' time series that exhibits patterns and is not a random white noise [3].

### 4.1.1 Stationarity

A time-series is stationary if its properties do not depend on the time at which the series is observed [29].

In the non-stationary time-series in Figure 4.1 there is a clear downward trend between $t = 0$ and $t = 900$. Therefore the mean is not constant over time. Whereas in the stationary time-series there are fluctuations, but the mean is always the same. The stationary time-series is much easier to model, which is why stationarity is a common assumption in time-series analysis and it is also an assumption for the ARIMA model.

### 4.1.2 Auto Regressive Model

The Auto Regressive Model, AR(p), is a regression model where the dependent variable depends on past values of itself [9].

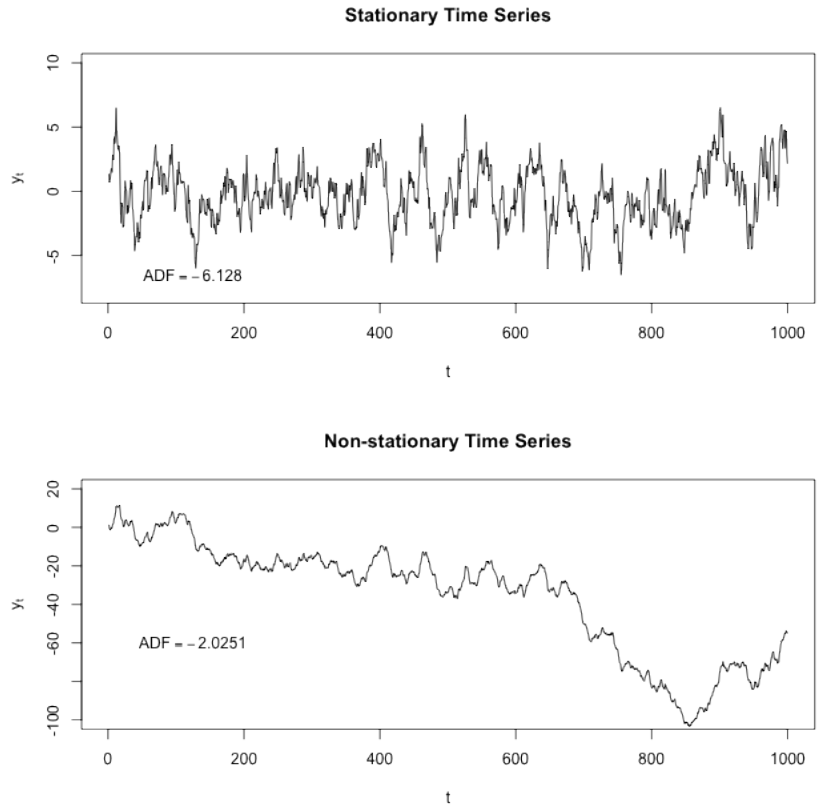



Figure 4.1: Time series generated by a stationary (top) and a non-stationary (bottom) process. *Source: [33]*

$p$ in the ARIMA model always refers to the order of the AR term. It denotes the lag, i.e. how many past values to consider. The following equation shows how $X_t$ can be written as a function of the lags of $X_t$ [3].

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t \tag{4.1}$$

where $c$ is the intercept term, $X_{t-i}$ are the lags of $X_t$, $\phi_i$ are the coefficients of the lags, and $\epsilon_t$ is white noise.

A well-known example of this model is the Simple Moving Average Model, where the coefficients $\phi_i$ are specified:

$$X_t = \frac{X_{t-1} + X_{t-2} + ... + X_{t-p}}{p} \tag{4.2}$$

Thus, $\phi_i = \frac{1}{p}$ for all $i$. In this case, the model does not have to be fitted to the data, because the coefficients are already pre-defined. If the coefficients, $\phi_i$, are not given, then AR(p) finds the coefficients with the best fit.

Figure 4.2 shows an example of the simple moving average model on a respiratory signal. The red lines indicate the values given by calculating the simple moving average with a lag of 10 and the blue line indicates the original signal.

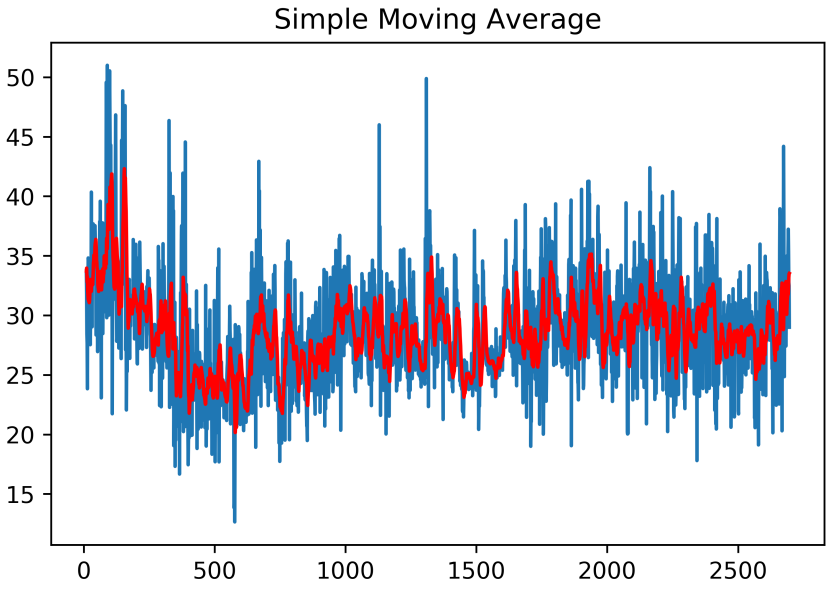



Figure 4.2: Simple Moving Average Example

### 4.1.3 Moving Average Model

The Moving Average Model, MA(q), is a regression model on past error terms, also called "lagged error regressive". The $q$ parameter in ARIMA always refers to the order of MA. The following function shows how $X_t$ depends on the lagged forecast errors.

$$X_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} \tag{4.3}$$

Here, $\mu$ is the intercept term and $\theta_i$ the coefficients.

### 4.1.4 Autoregressive Moving Average Model

The Autoregressive Moving Average Model, ARMA(p,q), is a merger between AR(p) and MA(q).

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-1} + \sum_{i=1}^{q} \theta_i \epsilon_{t-1} \tag{4.4}$$

$c$ is the intercept term, $\phi_i$ are the coefficients of the lagged $X's$ and $\theta_i$ are the coefficients of the lagged error terms, $\epsilon_{t-1}$.

### 4.1.5 Autoregressive Integrated Moving Average Model

The Autoregressive Integrated Moving Average Model, ARIMA(p,d,q), is similar to ARMA with an additional parameter $d$ for the degree of differencing. Differencing means taking the difference of the current observation to the last observation. This procedure eliminates trend and seasonality and stabilizes the mean of the time-series making it stationary.

The equation is constructed as follows:

$$\begin{aligned}
&\text{If } d = 0 : x_t = X_t \\
&\text{If } d = 1 : x_t = X_t - X_{t-1} \\
&\text{If } d = 2 : x_t = (X_t - X_{t-1}) - (X_{t-1} - X_{t-2})
\end{aligned}$$

Depending on the parameter $d$ in ARIMA(p,d,q) the differencing is done as shown above, and for each t, a new $x_t$ is created. The new $x_t$'s replace the original $X_t$'s in the ARMA(p,q) model. The ARIMA forecasting equation is:

$$x_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i x_{t-1} + \sum_{i=1}^{q} \theta_i \epsilon_{t-1} \tag{4.5}$$

which is identical to ARMA(p,q) except that the variables $X_t$ are differenced according to the parameter $d$.

## 4.2 Implementation

All the models were implemented using Jupyter Notebook. A first analysis was done on the MIMIC III Database, but finally the PhysioNet Challenge 2015 Database was used for training and testing because it contained more information on whether or not the alarms were true or false.

All the information regarding the MIMIC III Database can be found in Appendix A. Some helpful information from the analysis on the MIMIC III Database could be adopted including the GridSearch parameters.

#### Signal Preprocessing

The following preprocessing methods were tested for each signal to determine which methods yield the best results.

- Butterworth Filter
- Pacemaker Spike removal through interpolation
- Median Filter
- Downsampling

These preprocessing techniques were introduced in chapter 3. All the parameters for the techniques were adopted from the participants in the PhysioNet Challenge 2015. If more than one team used that specific technique, the parameters from the higher scoring team were used. Depending on which alarm type was raised and which signal is being processed, the set of optimal preprocessing strategies differed. In all methods, each processed signal was modelled with an ARIMA model using a rolling forecast and forecasting only one sample at a time. The parameters of the ARIMA model were found using GridSearch. The absolute errors of the model for each signal was saved.

### Training and Testing

The PhysioNet Challenge 2015 Dataset consists of 750 records. Those records were split into a train set and a test set with the last 20% of the records in the test set. The models were scored using Equation 2.1 in Chapter 2.1, which is the same metric as in the PhysioNet Challenge 2015.

The score is similar to the accuracy score except that false negatives are given more weight. The test set used in the challenge was not available, therefore the comparison to the participant's scores was limited.

### Baseline Model

Because it is safer to find all the true alarms and classify some false alarms as true, than the other way around, a simple Baseline model was evaluated, where each prediction is always "true alarm". The Baseline model scored 48.67 on the test set. A useful model should therefore score at least as well as this simple Baseline model. The following methods were implemented on the PhysioNet Challenge 2015 Database and the results are compared to the Baseline model.

## 4.3 Methods and Results

Several methods using ARIMA were tested to evaluate whether ARIMA could potentially be used for medical data analysis and alarm predictions. The initial idea was to model the signals using ARIMA and track the error between the ARIMA prediction and the true value. If the error suddenly gets bigger, then an alarm should be raised. Because this data contains real and false alarms, and the goal is not to raise an alarm at the correct time, but to classify whether the raised alarm was correctly raised or not, the initial idea had to be slightly changed. The most straightforward approach was to track the errors and if the errors became higher closer to the alarm, then a "true alarm" should be predicted, and else a "false alarm". However, this method did not seem to work at all, and after some analysis it turned out that the ARIMA errors were on average larger for false alarms than for true alarms. Hence, it made more sense, and resulted in better scores, to predict a "false alarm", if the ARIMA errors became higher closer to the alarm, and else predict a "true alarm". This idea resulted in several different methods, which differ slightly, but are all based on this intuition.

**Method 1** The two ECG leads II and V were preprocessed with a Butterworth Filter and Pacemaker Spikes were removed following the approach proposed by [18]. The ABP, PLETH and RESP signals were downsampled. Then the processed signals were modeled with an ARIMA model and the absolute errors saved. The errors were then fitted using a linear regression model and the slope of the linear regression determined whether to predict a true alarm or a false alarm. When the regression line was upwards sloping, i.e. the errors were higher right before the alarm was raised than 5 minutes earlier, a false alarm (= 0) was predicted. When

the regression line was downwards sloping a true alarm (= 1) was predicted.

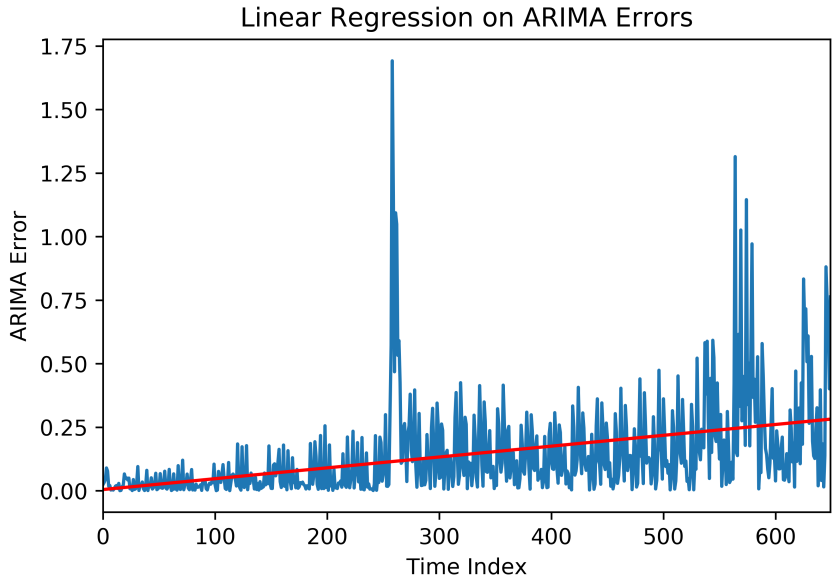



Figure 4.3: Linear Regression Record a186s

Figure 4.3 shows an example of this method on record a186s which had a false asystole alarm. The ECG II signal was modelled with ARIMA und the blue lines show the absolute errors of the rolling ARIMA forecasts. The red regression line is clearly upwards sloping and this model would therefore predict "false alarm" for this signal.

This resulted in a set of binary predictions for each record. For the final prediction the minimum was taken out of all predictions. In other words, if at least one signal resulted in a "false alarm" prediction, the overall prediction for that patient was "false alarm". Other methods including taking the mean or the max were tested but this method resulted in the best training score.

After training this method on the training set and tuning all the parameters, the method was tested on the test set. Figure 4.4 shows the resulting confusion matrices. Each alarm type was tested separately because the parameters differed greatly depending on the alarm. From top left to bottom the alarms are Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

The overall score on the test set resulted in 20.0, which is far below the Baseline. In Table 4.1 it can be seen that this model scores better than the Baseline for the Asystole and Ventricular Fibrillation alarms, but that other methods scored even better on those.

**Method 2** Instead of fitting the errors with a linear regression, the mean of the errors was compared to a predefined threshold. If the mean was above the threshold, the model predicted "false alarm", otherwise "true alarm". Each signal had its

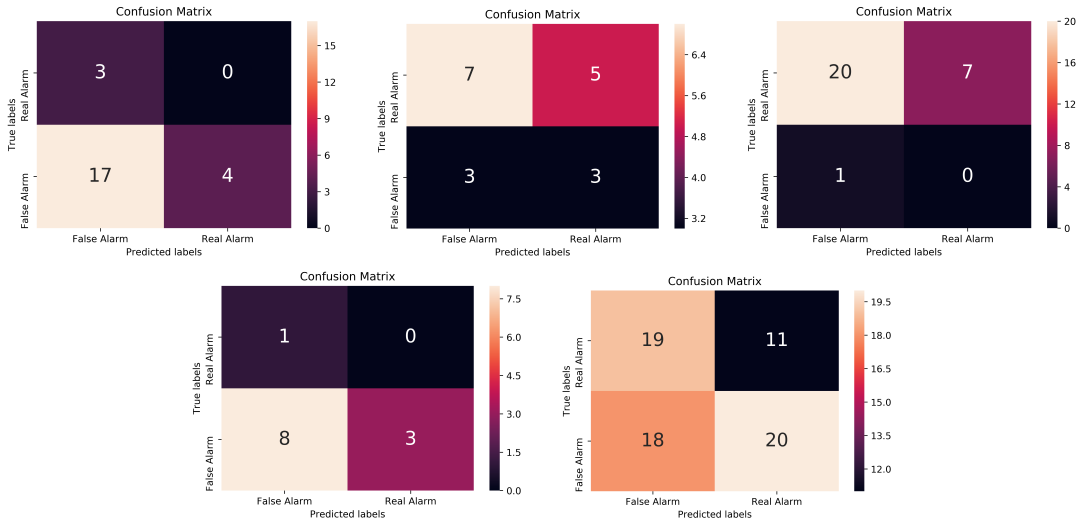



Figure 4.4: Confusion Matrix Method 1: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia

own threshold which were found by testing the model on the training set.

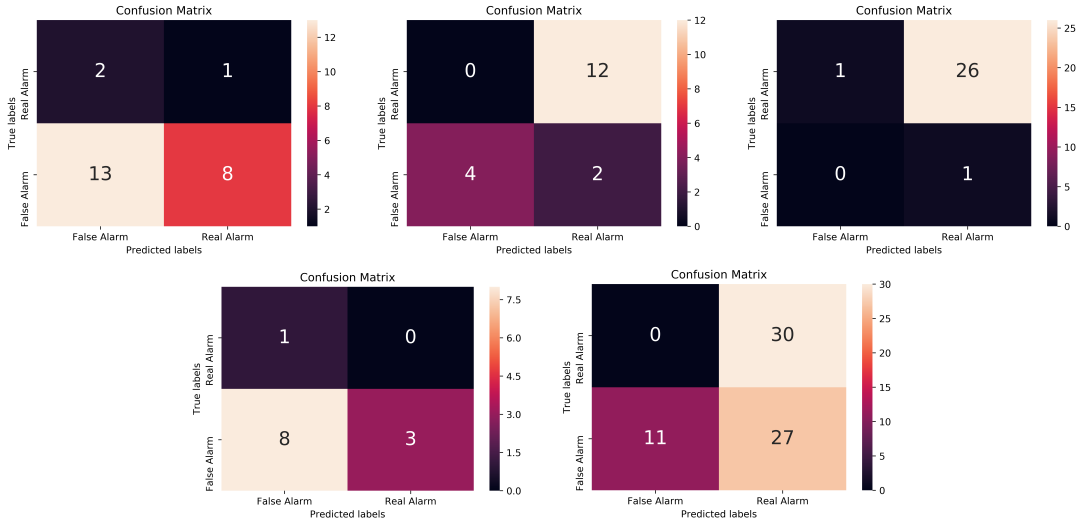



Figure 4.5: Confusion Matrix Method 2: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia

The overall score of method 2 resulted in 63.25. This is well above the Baseline. From Table 4.1 it can be seen that this method scored better than the Baseline for all alarms, except for the Extreme Tachycardia alarms. It scored especially well, compared to the other methods mentioned in this chapter, for the Extreme Bradycardia and Ventricular Tachycardia alarms. Most of the participants in the PhysioNet Challenge 2015 had the greatest difficulty classifying the Ventricular Tachycardia alarms. The winning team scored 75.07 on the hidden test set for that particular alarm which is below their overall score of 81.39 [30].

**Method 3** In this method the mean of the errors of the first $n$ seconds were compared to the mean of the errors of the last $n$ seconds. The idea is that the errors are likely to change a few seconds before the alarm is raised. The differences between the first $n$ seconds and last $n$ seconds were saved for each signal. The maximum difference per record was then compared to a predefined threshold and if it surpassed the threshold the model predicted "false alarm" and otherwise "true alarm". Both $n$ and the threshold were tuned based on the results of the training set.

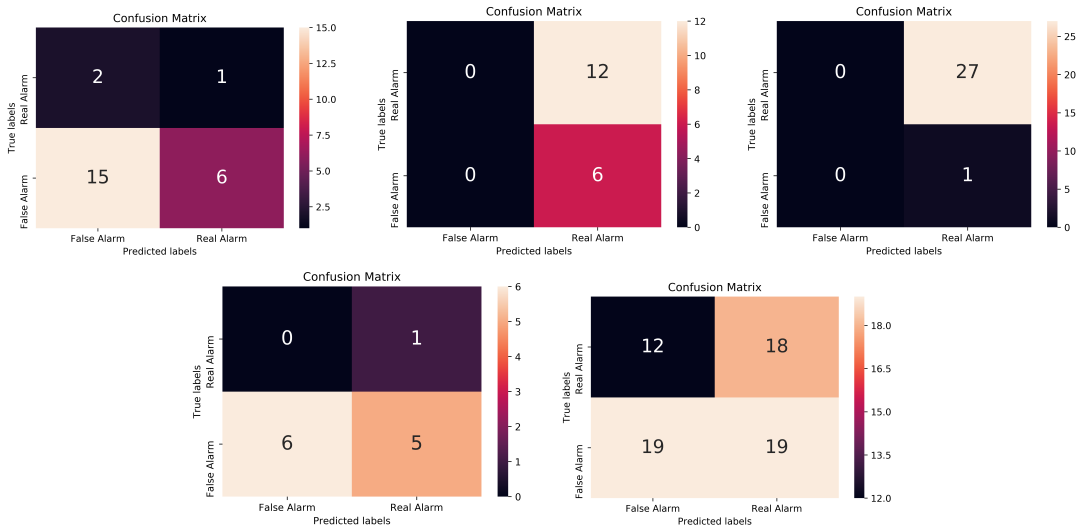



Figure 4.6: Confusion Matrix Method 3: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia

Figure 4.6 shows the confusion matrices for method 3. The overall achieved score is 48.06, which is slightly below the Baseline. Table 4.1 shows that it achieved better scores than the Baseline for all alarms, except the Ventricular Tachycardia alarms. It scored especially well on Ventricular Fibrillation.

**Method 4** In this method the maximum error (or 95th, 99th, ... quantile) of each signal is compared to the mean error of the signal. The idea is that if the alarm is caused by sudden noise, this might be much harder for the ARIMA model to forecast and this will result in a much higher maximum error compared to the mean than for a true alarm. The differences were saved for each signal and the mean of all differences computed per record and compared to a threshold.

Figure 4.7 shows the confusion matrix and the overall score of method 4 is 37.35 which is far below the Baseline. However, it scored better, or at least as good, on all alarms except Ventricular Tachycardia alarms.

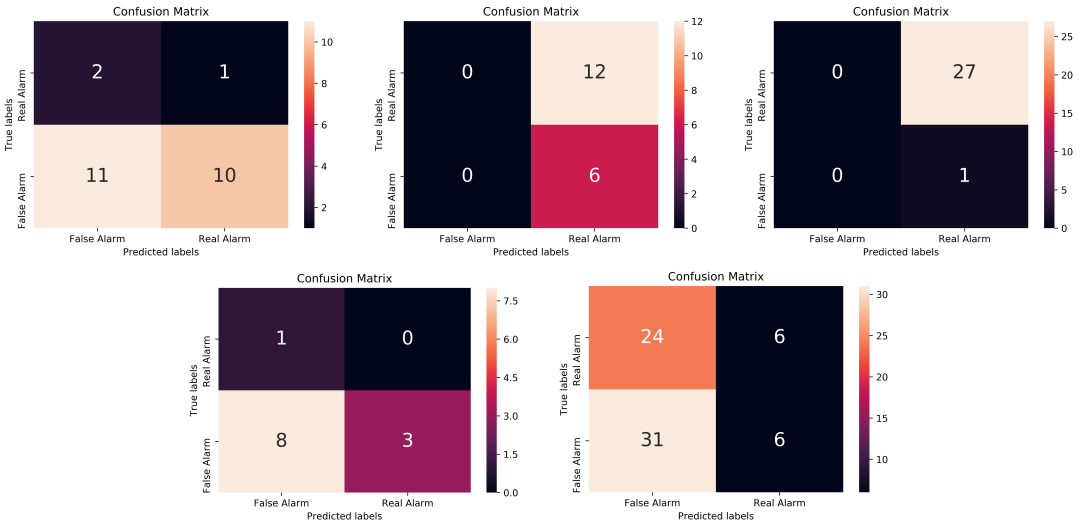



Figure 4.7: Confusion Matrix Method 4: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia

| | **Alarm Types** | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| Baseline | 12.5 | 66.67 | 96.43 | 8.33 | 44.78 |
| Method 1 | 47.22 | 17.39 | 7.41 | 50 | 20.14 |
| Method 2 | 43.75 | 88.89 | 81.25 | 50 | 60.29 |
| Method 3 | 50 | 66.67 | 96.43 | 58.33 | 31.9 |
| Method 4 | 37.5 | 66.67 | 96.43 | 50 | 22.7 |

Table 4.1: Comparison of Test Scores

## 4.4 Discussion and Future Work

In general, none of the methods above performed very well. However, a few methods performed better than others and through the entire process several interesting insights were found.

Interestingly, ARIMA performs on average worse when the alarms were false. This can be explained by the reasons why a false alarm occurred. Unfortunately, Physionet's Challenge 2015 does not provide a lot of explanation on how the alarms in their database were raised. However, from the participant's reports it is clear that the problem lies in too much noise and bad quality signals. Because noises can occur abruptly with no indication, ARIMA can not adapt quick enough and the errors are therefore larger. If it is a true alarm, the signals might possibly indicate this earlier by certain changes and the ARIMA model can adapt to the new signal.

Tables 4.2, 4.3, and 4.4 show the differences between mean statistics of the median, mean and max ARIMA errors of false alarms and true alarms for five different

| | Median | |
|---|---|---|
| | False Alarms | True Alarms |
| II | 0.16 | 0.09 |
| V | 0.18 | 0.09 |
| PLETH | 0.25 | 0.15 |
| RESP | 0.13 | 0.14 |
| ABP | 8.06 | 8.65 |

Table 4.2: Median of ARIMA errors

| | Mean | |
|---|---|---|
| | False Alarms | True Alarms |
| II | 966.71 | 0.15 |
| V | 1156.77 | 0.15 |
| PLETH | 126.29 | 75.76 |
| RESP | 475.16 | 0.17 |
| ABP | 6415.83 | 11.52 |

Table 4.3: Mean of ARIMA errors

signals. The median ARIMA error is bigger for false alarms for the signals II, V and PLETH. However, it is smaller for the signals RESP and ABP. This again shows that the ARIMA errors can vary greatly between different signals. The mean and max ARIMA errors are clearly much larger for false alarms than true alarms over all signals. There are a few very large outliers in the false alarms, which contribute mainly to those large errors in the mean. For example, the mean errors of the signal II for false alarms is 966.71. However, most errors are between 0 and 1. The maximum is 368'451, which of course pulls the mean up by quite a lot.

An investigation of these outliers reveals that they occur by sudden spikes in the signal, which an ARIMA model cannot predict. Figure 4.8 shows an example of the respiration signal for record a378s. The values are always around -8.18 and then suddenly there are a few huge jumps. This causes large ARIMA errors at the times of the jumps, which then results in a large mean ARIMA error for that record.

Hence, all the mean ARIMA errors above 100 were considered outliers and deleted. This resulted in 11 deleted values overall.

Table 4.5 shows the mean ARIMA errors after removing the very large outliers. The differences between false alarm and true alarm are much smaller now, but for the signals II, V and PLETH, these statistics clearly show that there is indeed a difference in the ARIMA error depending whether the alarm was true or false. However, the difference does not seem large enough to build a strong model using only the information of the ARIMA errors.

Table 4.6 gives further evidence why even though the mean ARIMA errors are clearly larger for false alarms than true alarms it is hard to build a strong model. The problem is that the variance of the means over all false alarm records are quite

| | Max | |
|---|---|---|
| | False Alarms | True Alarms |
| II | 119950.24 | 1.35 |
| V | 165407.41 | 1.24 |
| PLETH | 16991.68 | 13470.11 |
| RESP | 50354.31 | 0.81 |
| ABP | 1053309.68 | 59.51 |

Table 4.4: Max of ARIMA errors

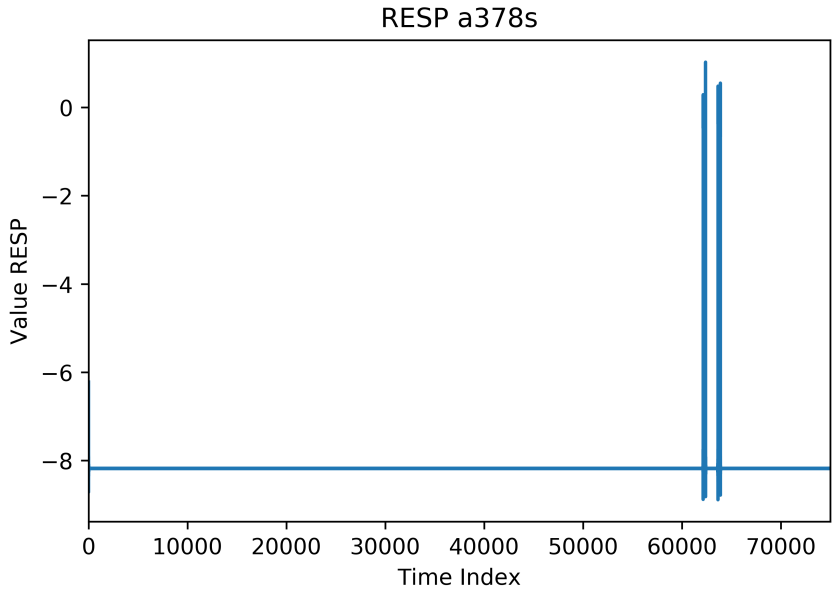



Figure 4.8: Example of a signal causing a very large ARIMA error

large. For each signal the variance of the mean errors of false alarms is larger than the variance of the mean errors of true alarms. This means that even though the mean error is larger over all false alarm records, there are still a lot of false alarm records which will result in a small mean error.

In Figure 4.9 the same maximum outliers were removed to show a reasonable distribution plot of the ARIMA errors and to compare the distribution between false alarms and true alarms. The distributions show a larger variance for false alarms than true alarms. The peak of the distributions, however, are very similar. It would be more preferable to obtain a model that produces errors with distribution peaks at different locations. Then a threshold function on the errors would result in a more distinct separation between false alarms and true alarms. Unfortunately, this was not achieved with ARIMA.

Four different methods using ARIMA were tested and analyzed. All methods used the ARIMA errors mentioned above. The differences between the methods lies in the algorithm for predicting false and true alarms using those errors.

Method 1 scored the worst compared to the other methods. The confusion ma-

| | Mean ARIMA Error | |
|---|---|---|
| | False Alarms | True Alarms |
| II | 0.28 (2 largest means removed) | 0.15 |
| V | 0.29 (2 largest means removed) | 0.15 |
| PLETH | 0.43 (4 largest means removed) | 0.19 (1 largest mean removed) |
| RESP | 0.17 (1 largest mean removed) | 0.19 |
| ABP | 11.22 (1 largest mean removed) | 11.52 |

Table 4.5: Mean ARIMA Errors without Outliers

| | Variance of Mean ARIMA Error | |
|---|---|---|
| | False Alarms | True Alarms |
| II | 0.39 (2 largest means removed) | 0.06 |
| V | 0.14 (2 largest means removed) | 0.01 |
| PLETH | 2.23 (4 largest means removed) | 0.04 (1 largest mean removed) |
| RESP | 0.08 (1 largest mean removed) | 0.03 |
| ABP | 55.88 (1 largest mean removed) | 36.88 |

Table 4.6: Variance of Mean ARIMA Errors over all Records

trices in Figure 4.4 show that the amount of False Negatives is a problem. False Negatives should be very small because it has a larger weight in the score metric. This means that the regression line was upwards sloping even though the alarm was true. Thus, the ARIMA error became on average worse the closer the alarm. Compared to the other methods, this method requires not a lot of parameter tuning. Since the model predicts "false alarm" for any upwards sloping regression line, even if it is almost horizontal, it predicts a "false alarm" too quickly. Further analysis could be done on whether the degree of the slope could be included as a threshold, i.e. only predict "false alarm" if the degree of the regression slope is larger than x, but this was not done because other methods showed more potential.

Method 2 is the only one that scored better than the Baseline. This method requires a lot of parameters to be tuned. Not only does each alarm type have its own threshold that must be tuned, but also each signal type has a different threshold. Tuning a lot of parameters is very difficult and time consuming and the risk of

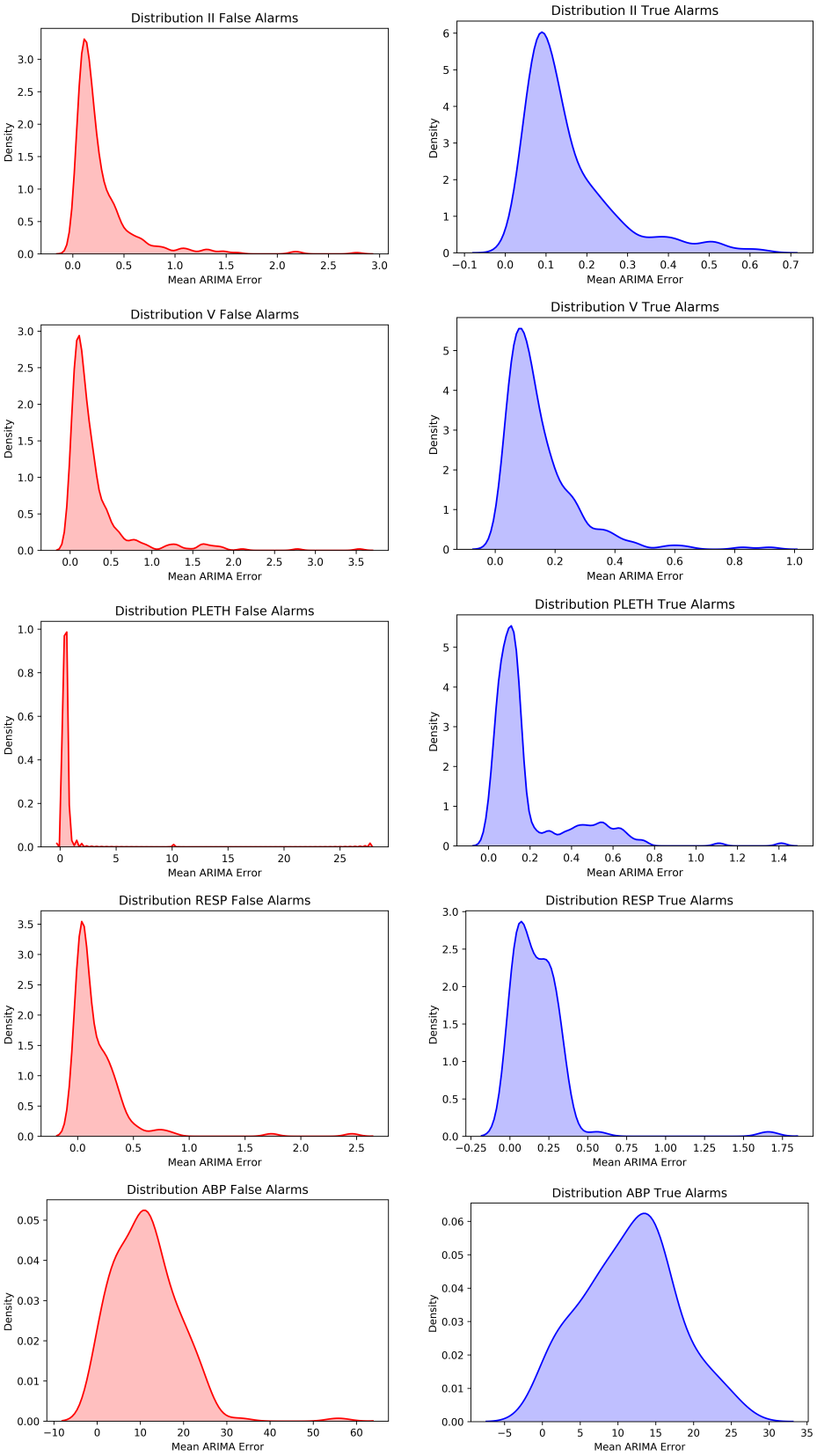



Figure 4.9: Distribution of Mean ARIMA Errors (After removing outliers)

overfitting to the training set increases. However, considering the results of these methods, it makes sense to view each signal independently, since this method performed much better than the other methods that tried to generalize all the signals and simplify the model.

Method 3 did not score better than the Baseline. However, it scored better on Asystole, Tachycardia and Flutter alarms compared to the other models. As already mentioned, on average the mean ARIMA error becomes higher before the alarm occurs if it is a false alarm. Unfortunately this effect is not large enough to build a strong model.

Method 4 scored lower than the Baseline. It scored well on Tachycardia alarms, but this is not very meaningful because the sample set for Tachycardia alarms was very unbalanced with 1 false alarm and 27 real alarms in the test set. There exist several methods to deal with imbalanced datasets, including upsampling and downsampling which may improve the scores of these methods.

Additionally, methods 3 and 4 could probably be improved by setting different threshold for each signal type similar to method 2. But this requires a lot of parameter tuning and the model becomes more complex.

Since Method 2 performed best on Bradycardia and Ventricular Tachycardia alarms and Method 3 performed best on Asystole, Tachycardia and Flutter alarms, it makes sense to use different methods for each alarm type. In this case an overall score of 67.72 can be achieved, which improves the overall score of only using Method 2. Of course, ensembles may also improve the score, i.e. combining more than one method for each signal. There are a lot of possibilities of combining all the methods and creating new methods using ARIMA errors as a base.

During this process, a big concern was always the number of samples. In total there are 750 samples which is not a lot to train a model. The test set contains only 150 samples. Since each alarm type is trained and tested separately, the number of samples becomes even smaller. The test set for Flutter alarms, for example, only consist of 12 samples. With such a small test set it is hard to obtain meaningful results. A small change in parameters can create a huge change in the score. The risk of overfitting is very high and it is likely that the models score very differently on more or different test samples. One solution would be to tune the parameters using cross validation of the training set. Instead of tuning the parameters on the entire training set, the idea would be to tune them on different samples of the training set and finding the set of parameters that maximizes the mean score over all samples. This was not carried out due to time constraints and because the overall goal was not to create a strong model, but to analyze whether ARIMA could be efficiently used for medical alarm predictions.

Another concern was that depending on the parameters of the ARIMA model, it could be extremely slow. Therefore, the parameters chosen were not only based on the best results but also with which parameters the model performed reasonably fast.

Despite all the downsides of ARIMA and the unsuccessful models, there are several advantages of using ARIMA or a similar model. One large advantage is that ARIMA automatically normalizes the patients. Many alarm raising algorithms work by raising an alarm if the signal passes a pre-defined threshold. This does not

work well firstly because it raises many false alarms due to noise but also because each patient is different and there should not be one pre-defined threshold for all patients. ARIMA adjusts to each patient separately. By only analyzing how the ARIMA errors change over time and not the actual signal, a good model could then generalize over all patients, unlike a threshold model.

Therefore, even though none of the models performed well on its own, it might be a good idea to use statistics of the ARIMA errors for other machine learning models. For example, for each signal, the mean, max, median, etc. of the ARIMA errors could be saved as features and used in a neural network to predict whether or not it is a true or a false alarm. For future work, this may have potential and should be further tested.

# Chapter 5

# Correlation-based Models

In this chapter, we are going to describe the design strategies and implementation of the correlation-based model. This model does not use any known approaches for time series forecasting, but rather the idea is to use cross-correlation to see how promising an algorithm from scratch will be. Each record consists of several signals, thus the idea is to examine whether cross-correlation between the signals has an impact on the final predictions. The cross-correlation model is based on the Pearson correlation coefficient.

## 5.1 Pearson Correlation Coefficient

Often referred to as the Pearson R test, the Pearson correlation coefficient measures the relationship and strength between variables or in our case will be time series. To determine the relationship between two time series the coefficient value which ranges between -1 to +1 indicates the extent to which the series are linearly related. The value r=1 means that there is a perfect positive correlation and the value r=-1 means that there is a perfect negative correlation between the series.

$$\mathbf{r} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{5.1}$$

The equation for the correlation coefficient is depicted in 5.1 where $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ are series that can either be extracted from a column or a row depending on the model.

Several assumptions need to be taken into account when running the Pearson correlation:

1. First of all, the variables(series) in question should both be normally distributed, which indicates how the values of the variables(series) are scattered.

2. Pearson correlation does not expect significant outliers. Thus, if they exist they can have a large effect on the Pearson correlation coefficient.

3. Variables(series) should be nominal and continuous. In case the variables are ordinal, different approaches have to be taken; for example Spearman correlation.

4. The observations are paired observations. For every value in a series, there must be a corresponding value in the other series. For example, in our case, all the signals have the same number of values.

## 5.2 Design Strategies for Prediction Algorithms

The models we build are correlation dependent and therefore no other additional or implemented algorithms are used. With cross-correlation, we measured how similar two series or signals are related to each other. This is not limited to cross-correlation between signals but also cross-correlation within the same signal. Two different window splitting techniques are used, the tumbling window and the sliding window which can also be seen in Figure 5.1.

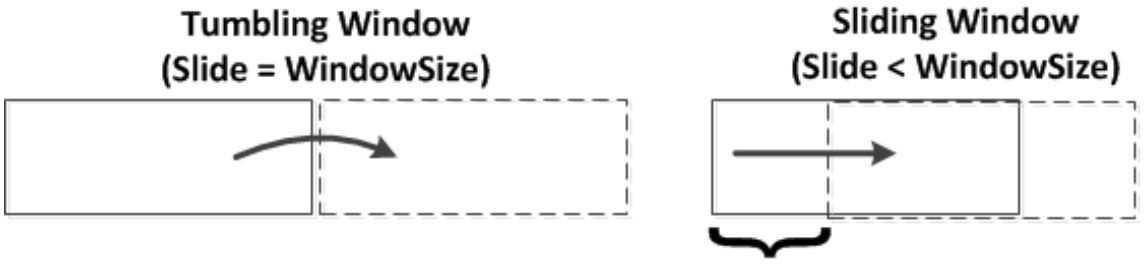



Figure 5.1: Windowing techniques

In the tumbling window, we extract windows or parts of the signals and correlate them with the proceeding windows. The chosen window, however, does not overlap with the next window, therefore each window is distinct.

On the other hand, we have also employed the sliding window technique, where the windows overlap while leaving one value behind when moving to the next window.

The main motivation for devising different strategies on building models using cross-correlation comes from the exploratory analysis. In the exploratory analysis, we have seen that ECG II and ECG V signals measure the heart activities from different angles. Therefore, these two signals represent the best fit between time series. Other signals appeared to be in different frequencies and very scattered.

In chapter 2 we also provided a table with the number of times each alarm appears in a record. Therefore the only signals that appear in 90 percent of cases are ECG II and ECG V followed by PLETH. By each alarm type, the ECG signals are affected. Therefore, ECG signals bring significant insights when an alarm is activated.

Since the cross correlation is applied to pairwise variables, three main strategies for building a model are used:

1. **Single-column based model**

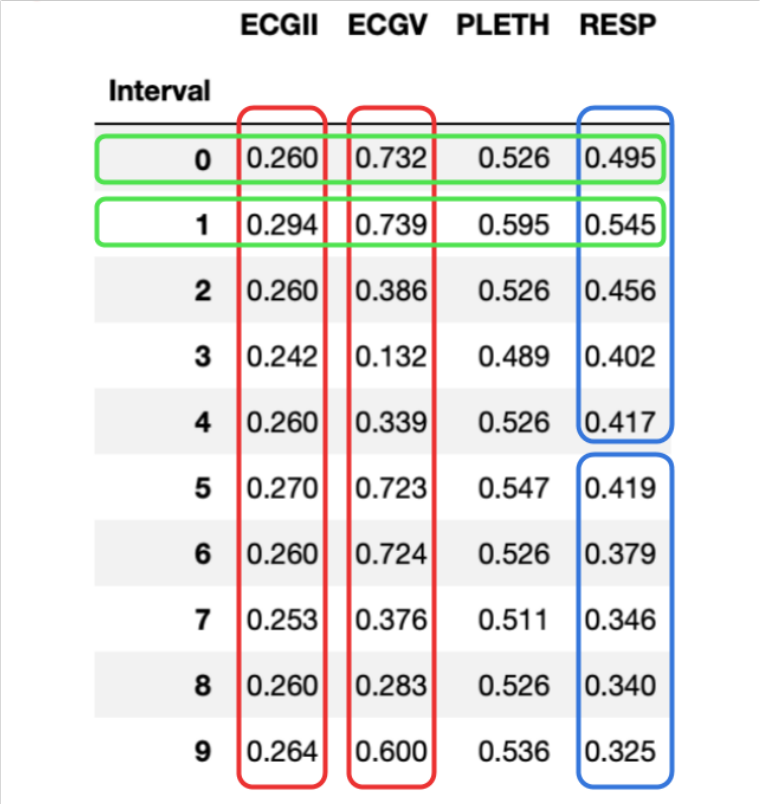
2. **Column-based model**

3. **Row-based model**

The three proposed models will predict alarms based on correlating windows within the same signal and correlation between different signals which are visualized on a snapshot of a record highlighted with colors representing the idea of the models in Figure 5.2. In the figure we can see 10 rows extracted from a record to elaborate the idea behind the models. The rows and columns selected do not manifest the final parameters such as window size and the pairwise signal comparison or the window splitting technique.

The first model, Single-column based model, highlighted in blue is based on correlating windows of one signal. In this case the window size is 5 and it wraps 5 rows. The two windows are correlated and stored for further processing. The model is applied to all signals by tuning the window size, window sliding technique and other parameters. The column-based model is highlighted in red in Figure 5.2. In this the case the window of ECG II signal is correlated with the corresponding window of ECG V. This is applied to all signals and tuned with other parameters to get the best result. The correlation is applied between columns and a threshold decides for the alarm to be true or false.

Similarly, in the row-based model elaborated in green in the above-mentioned figure, the windows cover the entire row including all columns. The window size in this case is 1 which will be calibrated while training the model. Eventually the correlation between rows is applied and an alarm is raised based on the threshold set. Further details will be explained in the following sections.



| Interval | ECGII | ECGV | PLETH | RESP |
|---|---|---|---|---|
| 0 | 0.260 | 0.732 | 0.526 | 0.495 |
| 1 | 0.294 | 0.739 | 0.595 | 0.545 |
| 2 | 0.260 | 0.386 | 0.526 | 0.456 |
| 3 | 0.242 | 0.132 | 0.489 | 0.402 |
| 4 | 0.260 | 0.339 | 0.526 | 0.417 |
| 5 | 0.270 | 0.723 | 0.547 | 0.419 |
| 6 | 0.260 | 0.724 | 0.526 | 0.379 |
| 7 | 0.253 | 0.376 | 0.511 | 0.346 |
| 8 | 0.260 | 0.283 | 0.526 | 0.340 |
| 9 | 0.264 | 0.600 | 0.536 | 0.325 |

Figure 5.2: Model visualisation: Single-column model, column-based and row-based in blue, red and green respectively.

## 5.3 Implementation and Results

The models were all implemented in Jupyter Notebook. For data analysis and model implementation mostly pandas library is used. Pandas offered a number of functions for correlation and sliding window. However most of the functions were implemented from scratch. Signal preprocessing functions mentioned in chapter 3 were also applied to see if that brings any benefits in the model prediction.

### Training and Testing

To test how well the models perform, the scoring function from PhysioNet Challenge 2015 was used for evaluation. The scoring function can be seen in chapter 4 which is a little different than the classification rate or accuracy because the false negatives are weighted more.

To visualize the performance of the models the confusion matrix will be provided for each alarm. However the confusion matrix does not represent the score of the function adopted from PhysioNet Challenge 2015. Therefore, the accuracy that can be extracted from the confusion matrix is not the same as the scoring function from PhysioNet.

### 5.3.1 Single-column Based Model

In the exploratory analysis, we emphasized the last 10 seconds of the records right before the alarm occurs. We compared the last window with the second last window and found a lot of insights. The main signals that measure heartbeats directly are ECG II and ECG V. Those signals also changed oscillation and frequency in the last 10 seconds. This means that the same signal in the last 10 seconds was less correlated with the preceding window of 10 seconds. The initial idea for building this model came right from there. Therefore in this model, we took a signal specifically ECG II because it offered a complete QRS complex and it is also present in almost all the records. The signals were split into windows of 1000 rows which correspond to 4 seconds of real-time. After that, each window was correlated with the proceeding window. The tumbling window technique was employed and no values were repeated. The reason for choosing this technique is because we needed to find a sharp difference in the correlation of windows. With a sliding window, we would not get the sharp difference as values repeatedly occur in many windows.

Therefore we pointed out that when correlating the second last window with the last window, little to no correlation was found. After that, we decided to find the discrete difference between the correlation coefficient received. From the current cell value, the previous row cell value was subtracted. When the difference is high that means the signal oscillation or frequency also has changed. That leads us to extract the max difference and based on that eventually an alarm is decided to be true or false. For this, we set a threshold and tested the model with various thresholds ranging from 0.3 to 1.0. In the next section, we will display the scores and the confusion matrix of the model based on the alarm type.

#### Results

The model is applied to each alarm type separately which leads to isolating the success of the model. In Figure 5.3 and 5.4 the confusion matrix is displayed for training and testing data respectively with the predicted truth values of each alarm type. We have also mentioned before the accuracy, precision or recall is not taken into account. However, the score of a model is based on the scoring function from the PhysioNet Challenge 2015. The scoring function puts more weight into the false negatives and considers them as life-threatening events. Due to that, the standard accuracy score may be slightly higher than the scoring function from Physionet.

The scores for each alarm type can be seen in Table 5.2. The model scores are high for two alarm types: Tachycardia and Ventricular Fibrillation or Flutter. Hence the model outperforms the baseline model for Asystole and Ventricular Fibrillation with high difference. Even though the score for Extreme Tachycardia is high still the baseline scores better. Likewise, the model is outperformed in the other two alarm types, Bradycardia and Ventricular Tachycardia.

The model has been trained with different parameters by taking into account the window size and the threshold. Therefore by calibrating the parameters, we stick to the model that resulted in the best score.

| | Alarm Types | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| Baseline | 12.5 | 66.67 | 96.43 | 8.33 | 44.78 |
| Method 1 | 46.67 | 27.73 | 92.86 | 67.24 | 40.64 |
| Method 2 | 44.25 | 39.08 | 92.86 | 62.12 | 42.38 |
| Method 3 | 28.26 | 20.00 | 20.00 | 40.00 | 38.66 |

Table 5.1: Training Scores

| | Alarm Types | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| Baseline | 12.5 | 66.67 | 96.43 | 8.33 | 44.78 |
| Method 1 | 47.22 | 18.42 | 96.43 | 91.67 | 20.11 |
| Method 2 | 58.33 | 38.46 | 96.43 | 68.75 | 21.2 |
| Method 3 | 40.62 | 28.95 | 22.22 | 50.00 | 30.17 |

Table 5.2: Testing Scores

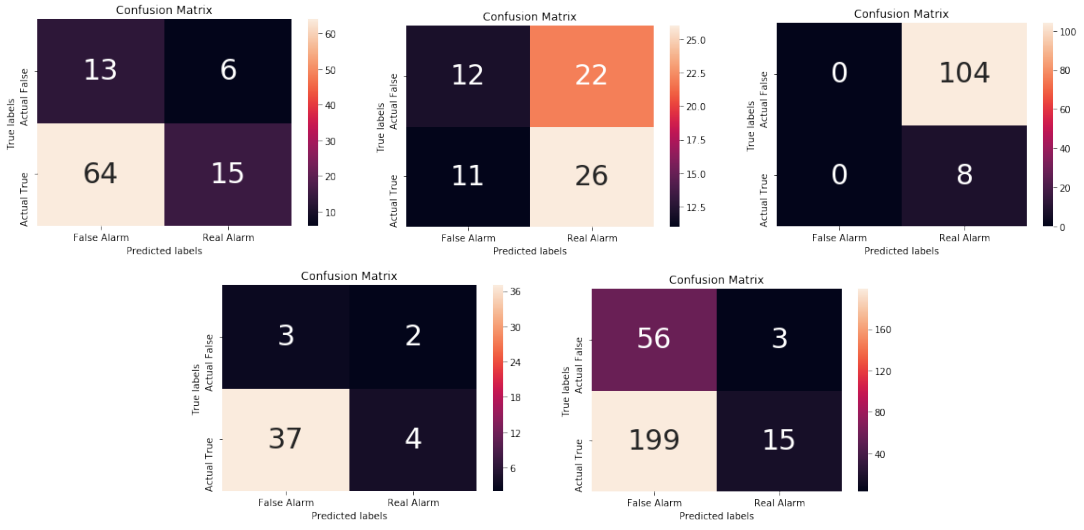



Figure 5.3: Confusion Matrix Method 1 trained data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

### 5.3.2 Column-based Model

In the exploratory analysis, we also discovered that the ECG II and ECG V signals are highly correlated with each other. This can be seen in the exploratory analysis

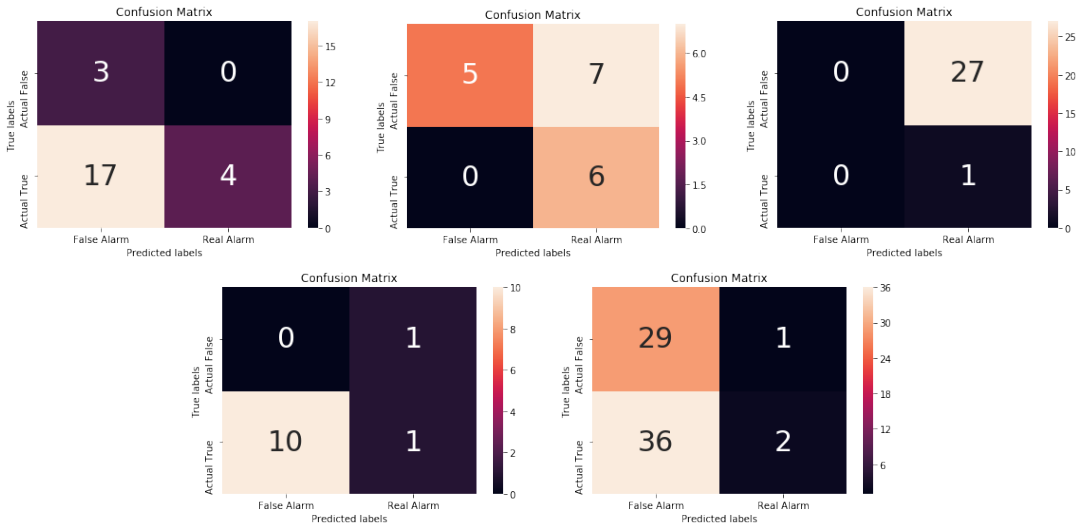



Figure 5.4: Confusion Matrix Method 1 test data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

chapter with red and blue lines. However, when an event happens before the alarm occurs the correlation between signals becomes weaker. Therefore the correlation between signals is no longer the same. For this reason, in this model, we decided to cross-correlate the signals between each other.

The model starts with correlating windows of the first column with the second, the second with the third, and so on until it reaches the last column. In the PhysioNet Challenge 2015 Database, the first and the second columns that appear in a record are often ECG II and ECG V and this is where the focus is. As mentioned before these two signals appear to correlate with each other and are present in almost all records whereas others are in a different frequency and oscillation and do not directly measure heartbeats. The best-fit windows size is 5 and the sliding window technique is used. So a window of one column is correlated with the window of the next column and the result is stored in a new column for further processing.

After that, the mean of the new column is computed. The newly created column is split into windows of 10 seconds or 2500 rows and the mean of each of these windows is computed. We know that an event happens in the last 10 seconds, so now we have the mean of the last 10 seconds and all the preceding windows before that. So the assumption is that the mean of the last window must be smaller than the mean of all the windows before it. This comes from the fact that the correlation of the last window of 10 seconds and the window before that is smaller. Therefore, in that case, the difference between the mean of the last window and the second last must be higher. So, in that case, we have a threshold and an alarm is raised when the difference is higher.

This is tested with several different threshold values with lower values being more tolerant and higher values more strict. Higher thresholds give a higher number of false alarms and lower thresholds give a higher number of true alarms. The model has been tested with different window sizes and thresholds. The most optimal model is chosen for testing.

**Results**

Accordingly, the model is applied to the five different types of alarms. In Figure 5.5 and Figure 5.4 we can see the confusion matrix of the training and testing results with the first row in the x-axis being correctly predicted alarms and the row on top with wrongly predicted alarms. From the confusion matrix, we can conclude that the higher the number of correctly predicted false alarms, the higher the score of the function designed by PhysioNet.

If we compare the scores shown in Table 5.2 with the confusion matrix we can see that even though the number of correctly predicted true alarms is high, if there are wrongly predicted false alarms then the score remains low. We can also say that the score of the trained data does not differ much except for Ventricular Tachycardia alarms and Ventricular Fibrillation or Flutter. The number of true and false alarms for each alarm type differs for example asystole alarms, we have roughly 80% false alarms and the opposite happens with Tachycardia which can be tricky to tune with the threshold and other parameters.

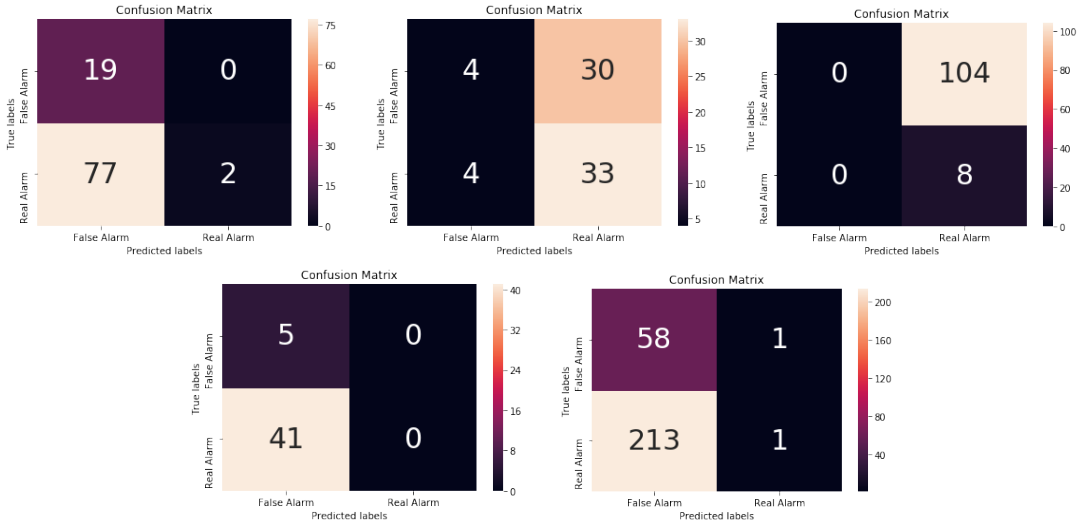



Figure 5.5: Confusion Matrix Method 2 trained data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

### 5.3.3 Row-based Model

In the row-based model as the name says, it is based on the correlation of rows. Similar to the column-based model where columns are correlated with each other, in this model instead the rows are correlated with each other.

So the idea is to correlate the first row with the second, the second with the third and so on. This is done by extracting the entire row as a series and then doing the same in the proceeding row and correlating the two series. The correlation of the rows will lead to a new column with the Pearson correlation coefficient. After that, we look at the discrete differences between the correlation values. This gives a lot of hints, if the signal is changing in its frequency or amplitude then the correlation coefficient must be changing and the difference will be higher. The newly created

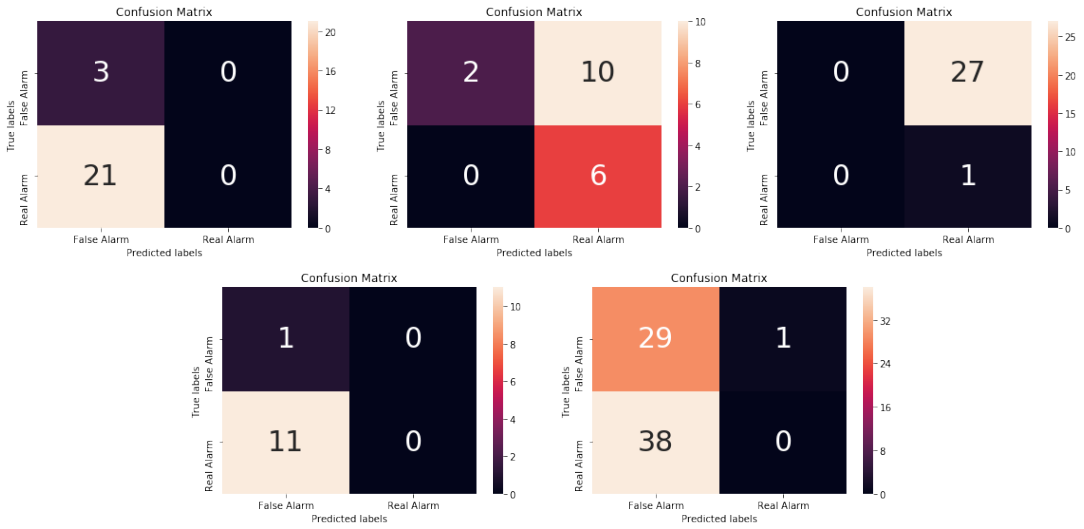



Figure 5.6: Confusion Matrix Method 2 test data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

column from the differences is split into several windows with a window size of 2000 rows or 8 seconds. The mean of each window is computed and then they are compared. If the mean of the last window is smaller than the mean of the first window, then a true alarm is raised, otherwise, a false alarm is raised.

In this model, the correlation between rows is done with correlating a single row with the next. So the window is of size one which is decided based on the idea to cover all the values but it leads to an algorithm with higher run-time complexity or quadratic time plus the additional number of rows which is 75000 rows.

In addition to the above approach of the row-based model, we tested a slightly different approach where instead of calculating the mean of the correlation coefficient we extracted the minimum and maximum values of the correlation coefficient. The alarms were decided true or false based on a threshold that was tuned between 0.3 and 0.9. However, this technique did not score well and was not very reliable.

### Results

The scores for this model can be seen in Table 5.2 and Table 5.1 for testing and training respectively. The tables reflect the scores for each alarm therefore the model performs differently in each alarm. The scores tend to be higher in the testing set compared to the scores in the training set. Therefore the scores in the testing set confirm that the model is stable even with fewer test data. The model has lower scores compared to the others, the reasons could be that the scores are generated from correlating rows of different signals. Each record does not necessarily contain the expected signals, therefore, different signals can appear and in different polarities. As we have mentioned earlier, ECGV signals sometimes appear in different patterns which means that sometimes the peaks are positive (upward) and negative (downward). This positive or negative deflection can appear in any ECG lead which depends on the depolarization which can spread toward the positive or negative pole of that lead.

The scores of this model are also reflected in the confusion matrix which is shown in Figure 5.7 and in Figure 5.8 for training and testing.

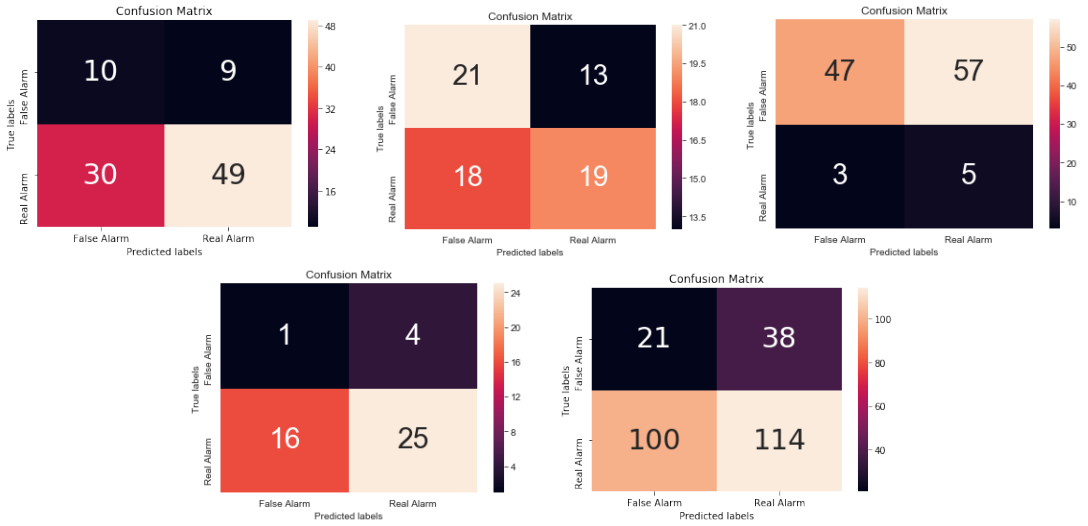



Figure 5.7: Confusion Matrix Method 3 trained data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

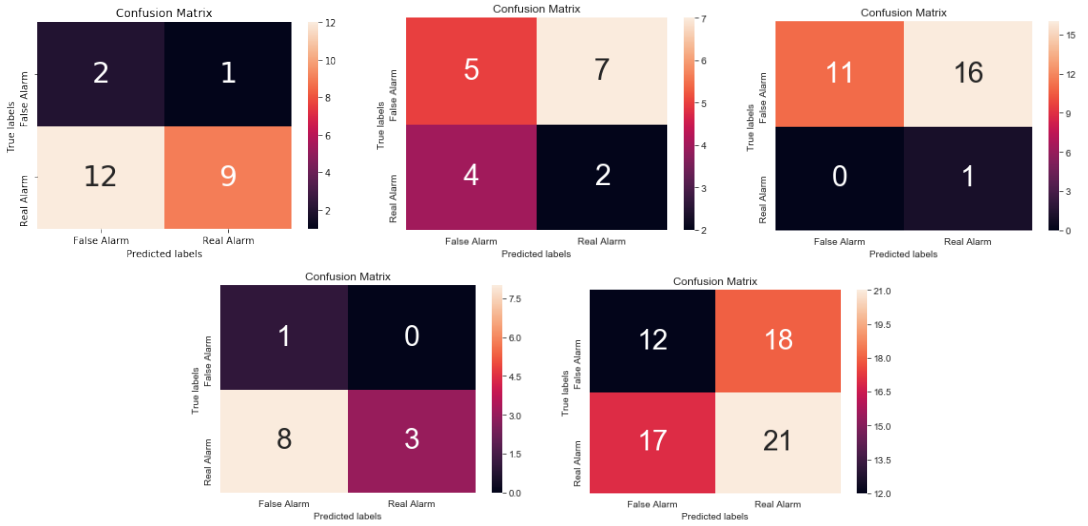



Figure 5.8: Confusion Matrix Method 3 test data: Asystole, Bradycardia, Tachycardia, Flutter and Ventricular Tachycardia.

## 5.4 Discussion and Future Work

The ideas behind the proposed models came from the fact that cross-correlation between different time series could lead to identifying changes in the signal over time. Those changes possibly reflect the health condition of a patient. Therefore the

scope of the models is bounded on the correlation of signals and no other additional models are incorporated.

As we have seen in the exploratory analysis the QRS wave pattern changes when the condition of the patient deteriorates. The PhysioNet Challenge 2015 Database came with several signals that correlated with each other either positively or negatively. Therefore in the exploratory analysis, it is also emphasized when the health condition of a patient deteriorates, also the signals change oscillation and frequency. Depending on the type of ECG V signal, the QRS wave, or the peak, sometimes appears upward or downward. Then there is either a high positive or negative correlation between ECG II and ECG V. Based on the insights from the exploratory analysis the three proposed models were trained and tested. The models have been trained and tested using the pre-processing functions mentioned in chapter 3, however that did not bring better scores. The sliding and tumbling window technique and other parameters have been calibrated to get the best results.

The scores for the first model or single-column based model can be seen in Table 5.2 and Table 5.1. The model scores well in Tachycardia and Ventricular Fibrillation but not in other alarms. The best score goes to Tachycardia alarms which are also reflected in the confusion matrix. We can see that in the confusion matrix of this alarm the number of false negatives is zero which means that the weight of an alarm not being raised as true has very high importance. However, in Bradycardia alarms, the score is quite low which is reflected also in the number of false negatives in the confusion matrix. In Figure 5.9 we can see the correlation values between windows of size 1000 rows, which is midway before extracting the absolute difference. On the left-hand side, we see the case of a true alarm and on the right side, we see the case of a false alarm. In the true alarm case, the values toward the end appear roughly close to zero which means that there is little to no correlation between windows in the last few seconds. In contrast, when the alarm is false the correlation appears to be high either positively or negatively.


Figure 5.9: Capture of correlation values from model 1

In the second model or column-based model, the results appear to be roughly similar to that of the first model. In comparison to the first model, this model performs better in Asystole alarms but weaker in Ventricular Fibrillation alarms. The scores are reflected in Table 5.2 and Table 5.1. Similarly in the confusion matrix when the number of false negatives is close to zero then the score is higher.

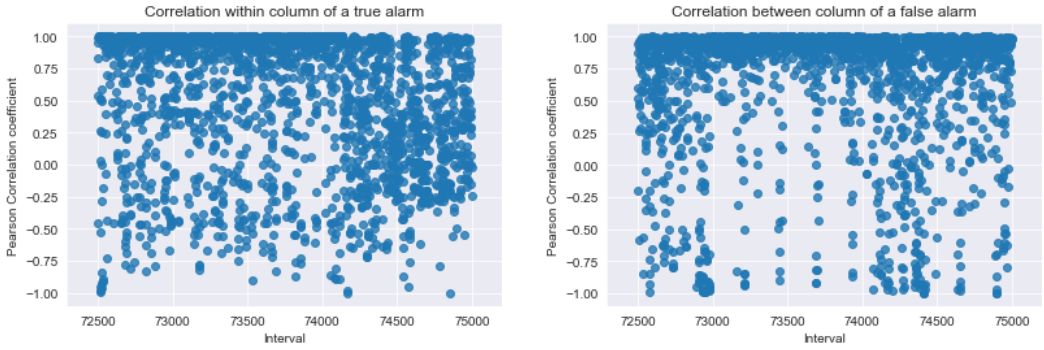



Figure 5.10: Capture of correlation values from model 2

In Figure 5.10 we see the correlation values between the first column and the second, ECG II and ECG V. In the true alarm the correlation values at the beginning lie mostly at the top or close to one, but towards the end, the correlation values are stretched toward the middle or close to zero which means that the correlation weakens and the number of heartbeats changes. On the right-hand side of Figure 5.10 the values lie mostly close to one except with some values that are scattered. The values that lie outside of the majority are few and still, it leaves no blank spaces near the correlation coefficient one.

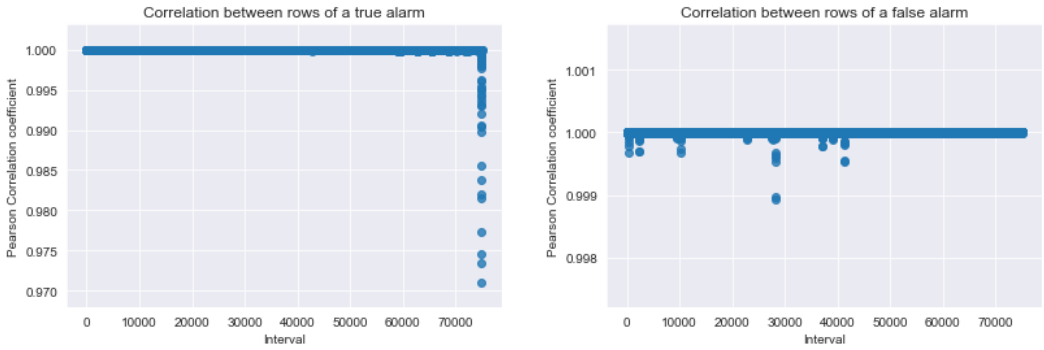



Figure 5.11: Capture of correlation values from model 3

The third model or row-based model has a lower performance compared to the other two. This can be because in this case all rows of different signals are taken into account and each record has different signals except the ECG II and ECG V which appear in almost all records.

In Figure 5.11 the correlation values between rows are shown before any other computation is done. In the true alarm throughout the whole record, rows are highly correlated with each other except when it reaches the last few seconds before the alarm happens. In the last few seconds, the values no longer correlate as strongly as before which shows that waves have changed the shape and a true alarm is raised. However, in the issued false alarm it can be seen that only some values lie outside the majority and not that far from the rest.

The models faced several challenges which we pinpointed through the training phase. The types of alarms in the PhysioNet Challenge 2015 are all heart-related

situations. First, a number of different signals are present in records but not all of them measure heartbeats. The two most important signals that directly measure the heartbeats are the ECG II and ECG V. The challenging part was that in the ECG V signals the QRS wave or the peak sometimes appear upwards and sometimes downwards. This means that there is sometimes either highly positive or negative correlation. In addition to that, it is not easy to detect accurate heart rate estimation only through correlation. Correlation can detect changes in heart rate but that makes it difficult to differentiate between low heart rate and high heart rate.

Another potential challenge for the row-based model could also be the algorithm time complexity when implementing it for real-time predictions. Its time complexity is quadratic plus the number of rows or 75000. This makes it slower than the other two models.

In general, the first two models performed better compared with the third model. A number of different parameters and strategies have been used to train the models. Some parameters performed better in a model but not in another. So they had to be calibrated for each model. Although the scores are not very high the models still perform well taking into account that the model is developed from scratch and no other known machine learning algorithms are used. The main idea behind the implemented models is to test how well correlation could support machine learning models for predicting time series data. We find that the proposed correlation models have a high potential in cases when detecting changes is needed.

We would like to propose a new approach to solving the high number of false alarms. Another potential algorithm for detecting heart issues could be detecting peaks. Each peak or QRS wave counts as one heartbeat. At the beginning we showed a graph on how the ECG signal looks like, therefore if we can locate the R wave from the QRS wave then we can calculate heart rate by measuring the R-R interval, which is the distance between two consecutive R waves. Since ECG is composed of a number of boxes and each box is 0.04 seconds, so counting the number of boxes between the QRS leads us to the heart rate [5]. Assume the number of boxes between the R-R interval is 5 then it makes 0.2 seconds, so 60 seconds divided by 0.2 brings 300 which means that the heart rate is 300 beats per minute. When there are 15 boxes between the R-R interval that means 0.6 seconds in between and in total it makes 100 beats per minute. This method could be very beneficial for the 4 alarm types, Asystole, Bradycardia, Tachycardia, and Ventricular Tachycardia.

This method should not be limited to only R wave, but we could take also the QRS wave but that will make it more difficult to detect each Q, R, and S waves. In the case of the Ventricular Fibrillation or Flutter, the QRS wave is no longer available, the rhythm is chaotic [10]. In this case, if for longer than 5 seconds there is no QRS wave then an alarm can be raised.

To materialize the proposed approach several ECG R-peak detection algorithms are available [6], some of which were used in the PhysioNet Challenge 2015.

To encapsulate, the correlation-based models proposed and implemented in this project showed that even algorithms developed from scratch that comes from new ideas can be successful and pave the way for discovering new approaches to solving a number of issues faced by the current predictive models.

# Chapter 6

# Cross Correlation Of Windows In The Same Signal Model

Correlating data between windows of the same time series has become even more critical to extract knowledge from time series data and explore correlations of windows of the same time series in nearly real time since the decisions whether to raise or not an alarm need to be made in real time as well. Subsequently, reducing the complexity of the model and increasing its efficiency were crucial to aim for real time decision making.

This model aimed to achieve a high false alarm suppressing ratio with a low true alarm suppressing ratio by making the most of the patterns that are found by correlating windows of the same time series. One of the attributes of different types of time series is the discovery of highly or slightly correlated time series windows having high. This model is based on the type of window correlation which is synchronous, meaning that: Given $N_a$ time series, a start time $t_a$ and a window size w,find,for each window W of size w,all pairs of windows $W_1$ , $W_2$. are highly correlated with each other. As we can see from Figure 6.1,we decided that our windows are fixed sized windows and not evolving or dynamic windows.

We tried to avoid continuously unloading previous values and loading new values thus spending processing power and risk losing our model performance in identifying patterns of correlations. Moreover, we mark important time periods when certain events happen such as sudden drops or high values of different medical signals as seen on Figure 6.2 and Figure 6.3 in which the red line indicates the location of the raised alarm on the Asystole alarm.

To summarize, there were consistent attempts to improve the performance of the model by focusing on the detection of patterns and features from the raw time series data.

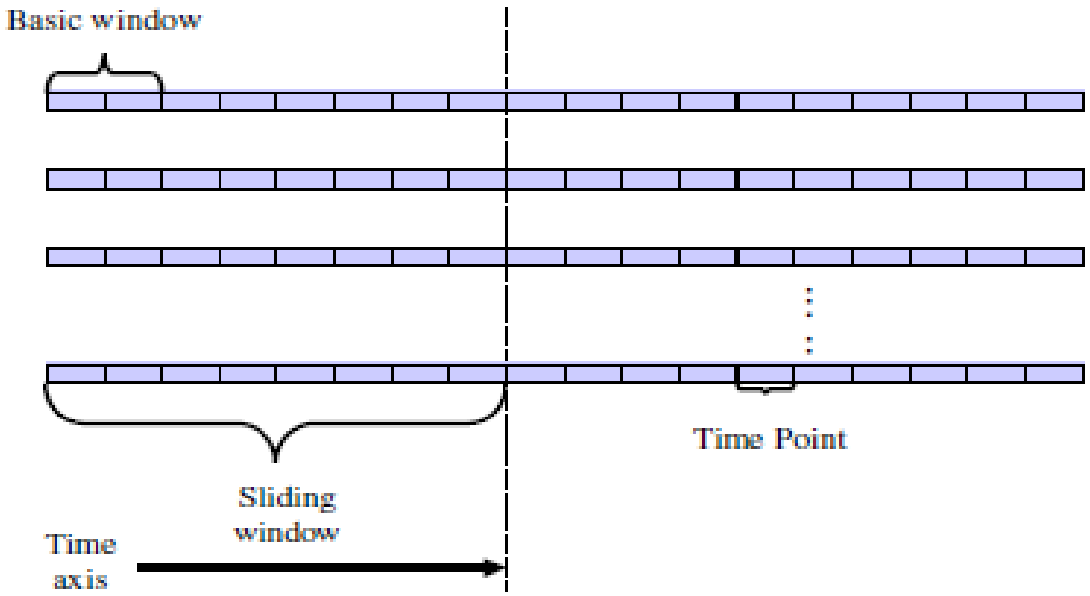



Figure 6.1: Fixed and dynamic window types

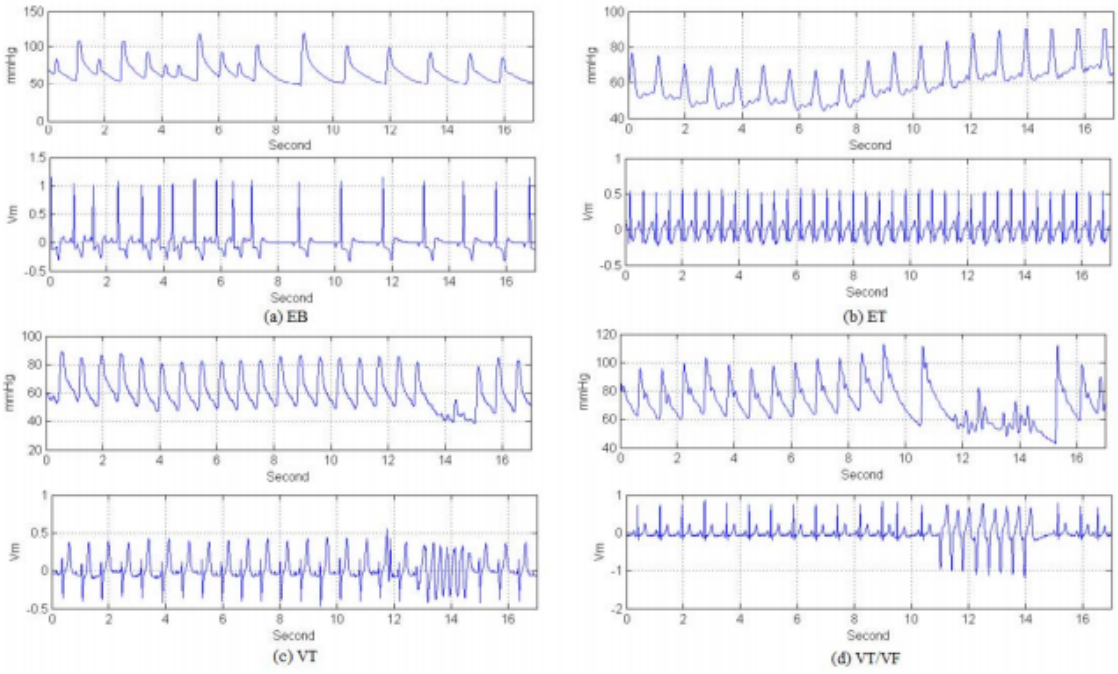



Figure 6.2: True and false alarms for different types of alarms

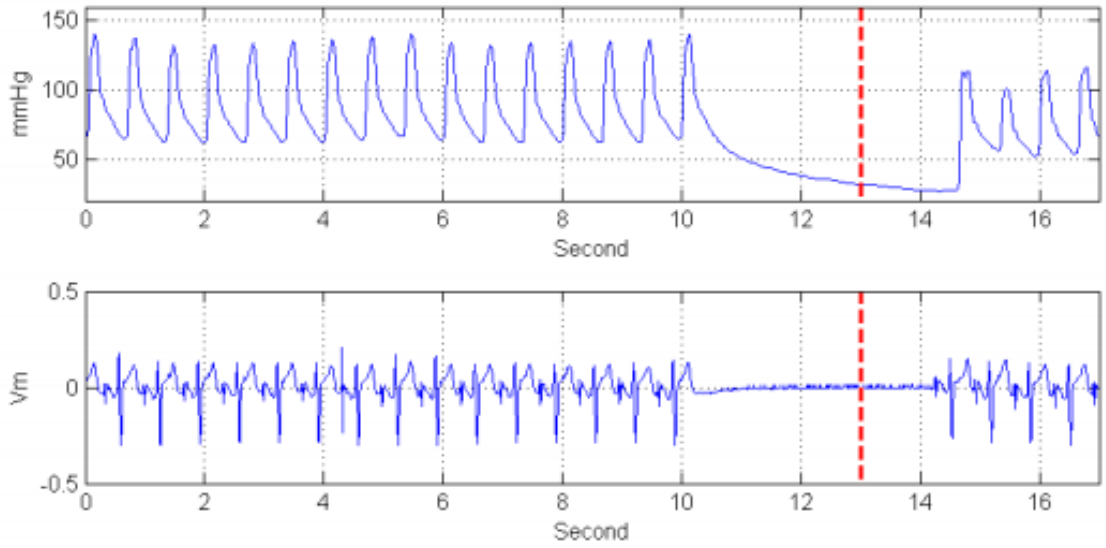



Figure 6.3: Location of alarm in Asystole signal

## 6.1 Implementation

The model was implemented using Jupyter Notebook and using different Python libraries ranging from Pandas to utilize handling and exploring the time series to Numpy and StatsModels. The dataset used for training and testing this model is the PhysioNet Challenge 2015 which was extensively discussed and described in the previous chapters.

Since the dataset consisted of only 750 records which were separated into 80% of it as training data and the remaining 20% as test data. It is worth to mention that if the size of the dataset would be larger then the results of our models would be more conclusive. Since the dataset from the PhysioNet Challenge 2015 was used, also the scoring metrics are quite close with the scoring function from PhysioNet Challenge 2015 as well. However, the model score is close to the accuracy score with the only downside that the one from PhysioNet puts more weight on false negatives due to the risks patients encounter as a consequence to false negative alarms and loss in efficiency of medical workers from false positive alarms.

The implementation of the model starts with the step of slicing the time series into windows of size 100 which are of fixed size and not evolving or dynamic windows, next they are correlated with each other producing a Pearson correlation coefficient. Each window is correlated with the following window and following a loop continuously for all pairs until we reach the end of the time series. The Pearson correlation coefficients are later used and compared on the threshold as a support for the decision if an alarm should be raised by the model or not.

The core part of model includes comparing and iterating through these window pairs and correlating pairs of windows which the correlation function performs on all of the values from one window with the values from the other respective window out of the pair. After each correlation is performed, the value is stored in an array and after each calculation of the correlation the new coefficient value is appended to the array.

Next the model iterates through all of the coefficients that were produced from the correlation of windows and correlates these values with each other. Since as observed during the exploratory data analysis many hints and signal patterns are found towards the end of the time series, after the function goes through all the records we get a final correlation coefficient from all of the window pair correlations with focus on the last windows in which the alarm happens and are found on the last 2500 rows of the time series. This final correlation coefficient is then compared with the threshold function and based on the difference with this threshold the model makes the classification for the alarm.

Finally, multiple window sizes were tested and different threshold values until the best results were obtained from the model as explained in detail in the following section.

## 6.2 Results

Since we have five different types of alarms the models were applied to each of them individually thus resulting in the real performance of the models on each alarm type. As a measure of performance the scoring function from the PhysioNet Challenge 2015 was used in these models as well, however there is a small difference between the standard scoring and the scoring function of PhysioNet Challenge 2015 since the latter puts more weight into the false negatives due to their effect in ICU as life-threating situations. The model scores were quite insightful since the models performed better in some alarm types and worse in others.

As it can be seen on Table **??** and Table **??**, the model scores account for two models; one with a window size of 50 and the other with a window size of 100. The first model scored better than the baseline in Asystole and Ventricular Fibrillation alarms and worse on Extreme Bradycardia, Extreme Tachycardia and Ventricular Tachycardia. Nevertheless, the second model with a window size of 100 showed slightly better scores on all types of alarms especially performs better than the baseline on Ventricular Tachycardia alarms. This improvement in performance is due to the larger windows not putting too much weight and relying on noise and bad quality signals, whereas smaller windows were affected more by noise.

Except for the models scores, the values for the false negatives and other values of confusion matrices of the window size 100 model for the five types of alarms are presented in Figure 6.4 and Figure 6.5. We can see that the model was more successful in finding false negatives in Asystole and Ventricular Tachycardia alarms however in other alarms such as Bradycardia, Tachycardia and Ventricular Fibrillation the model performed worse.

The weaker performance of the model in classifying Bradycardia alarms can also be explained by the fact that different Bradycardia alarms were found in the training set and different ones in the test set, nevertheless attempts were made to make these alarms consistent by focusing on the most reliable and error free signals containing Bradycardia alarms. To conclude, our model scores and confusion matrices represent the performance of our model but not the scoring function from PhysioNet Challenge 2015 which as mentioned before puts more weight on false negatives and is explained in detail in chapter 4.

| | **Alarm Types** | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| Baseline | 12.5 | 66.67 | 96.43 | 8.33 | 44.78 |
| Window size 50-Model | 46.29 | 29.08 | 83.58 | 51.33 | 40.77 |
| Window size 100-Model | 49.92 | 33.27 | 89.70 | 56.72 | 45.18 |

Table 6.1: Training Scores

| | Alarm Types | | | | |
|---|---|---|---|---|---|
| | Asystole | Extreme Bradycardia | Extreme Tachycardia | Ventricular Fibrillation | Ventricular Tachycardia |
| Baseline | 12.5 | 66.67 | 96.43 | 8.33 | 44.78 |
| Window size 50-Model | 47.63 | 38.30 | 85.26 | 54.70 | 43.12 |
| Window size 100-Model | 49.92 | 43.72 | 90.05 | 62.56 | 47.39 |

Table 6.2: Test Scores

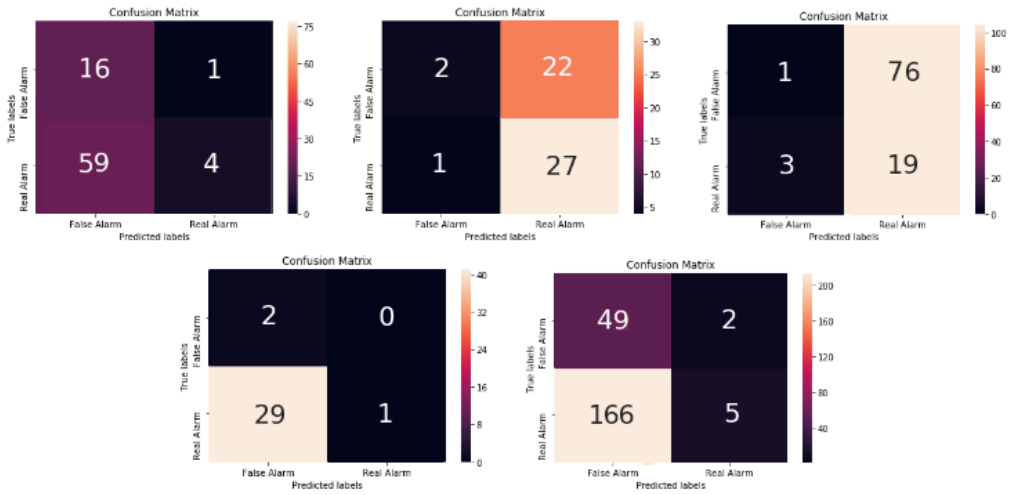



Figure 6.4: Confusion matrix of window size 100 model (trained data): Asystole, Bradycardia, Tachycardia, Ventricular Fibrillation and Ventricular Tachycardia

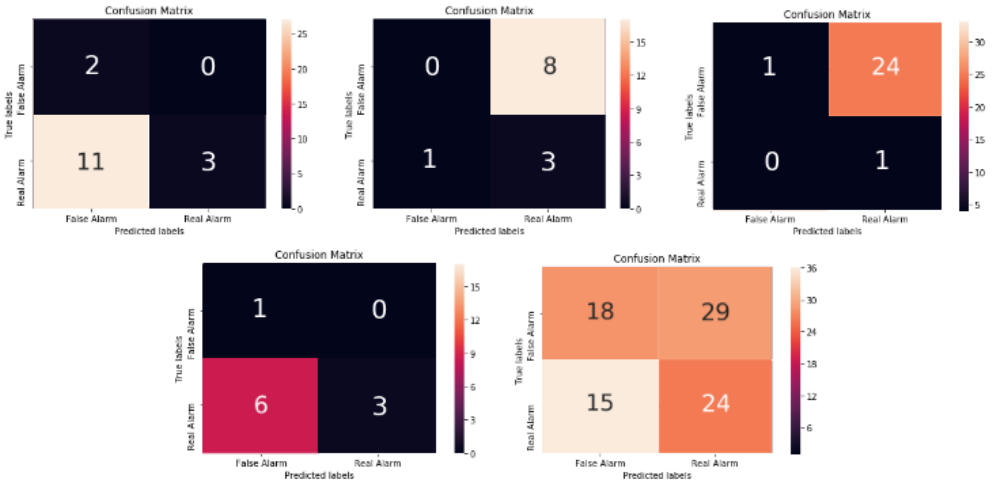



Figure 6.5: Confusion matrix of window size 100 model (test data): Asystole, Bradycardia, Tachycardia, Ventricular Fibrillation and Ventricular Tachycardia

## 6.3 Discussion and Future Work

Windowed correlation is quite an interesting approach because it indicates that parts of a signal which can be treated as two separate time series trend together over a certain duration of time. It can also significantly minimize the effect of noise and errors in the process of deciding if an alarm must be raised. Moreover, this type of correlation analysis is time-sensitive meaning that the time aspect is synchronous and not asynchronous (lagged). Taking that in consideration, window correlation is a common approach in detecting correlated features which in our case were medical metrics of the time series and monitor the change in correlation in windows of the same signal.

Through this model and its results the discovery of highly synchronous correlated windows was implemented and explored, thus this work is a step in this direction with the potential to be extended and improved in a few areas. Nevertheless, this study showed that correlation based machine learning models have potential in discovering changes and features of raw time series data.

New approaches that show potential in reducing false alarm rates include using Signal Quality Indices (SQIs) that are highly effective in detecting noise from normal signals and segmenting ECG signals which are clustered before finally learning the features from each cluster in a unsupervised way. Figure 6.6 shows the flowchart of the SQI algorithm which increases significantly the performance of machine learning models since it is highly effective in distinguishing noise (artifact) from normal signals. In [37] a Random Forest algorithm was applied after the SQI algorithm for feature selection, subsequently reducing the complexity of the model and increasing its performance.

Another potential approach which is similar to the above discussed model in terms of treating, handling and cutting signals is presented in [11]. The model implemented in this study segments the signals for input, clusters the segments to learn relationships among the segments and finally high-level features are extracted from the resulting segments. It is important to mention that signals are segmented by performing peak detection of the heartbeats, next they are fed to the K-means algorithm which clusters the segments which are most similar to each other. Figure 6.7 shows the results of their model, especially plot d) visualises how K-means algorithm clusters and detects QRS locations of the peaks in heartbeat signals.

To conclude, correlation based machine learning models performed good in detecting patterns of correlation and changes within the signals. They also show potential in the technique of using correlation as a tool to capture features of signals and learning them to predict alarms with low false negative rates. Moreover, recent work in this direction and new techniques of utilizing machine learning models are reducing the impact of noise in their models as the main factor of the high false alarm rates and at the same time reducing model complexity to make decisions about alarms in real time.

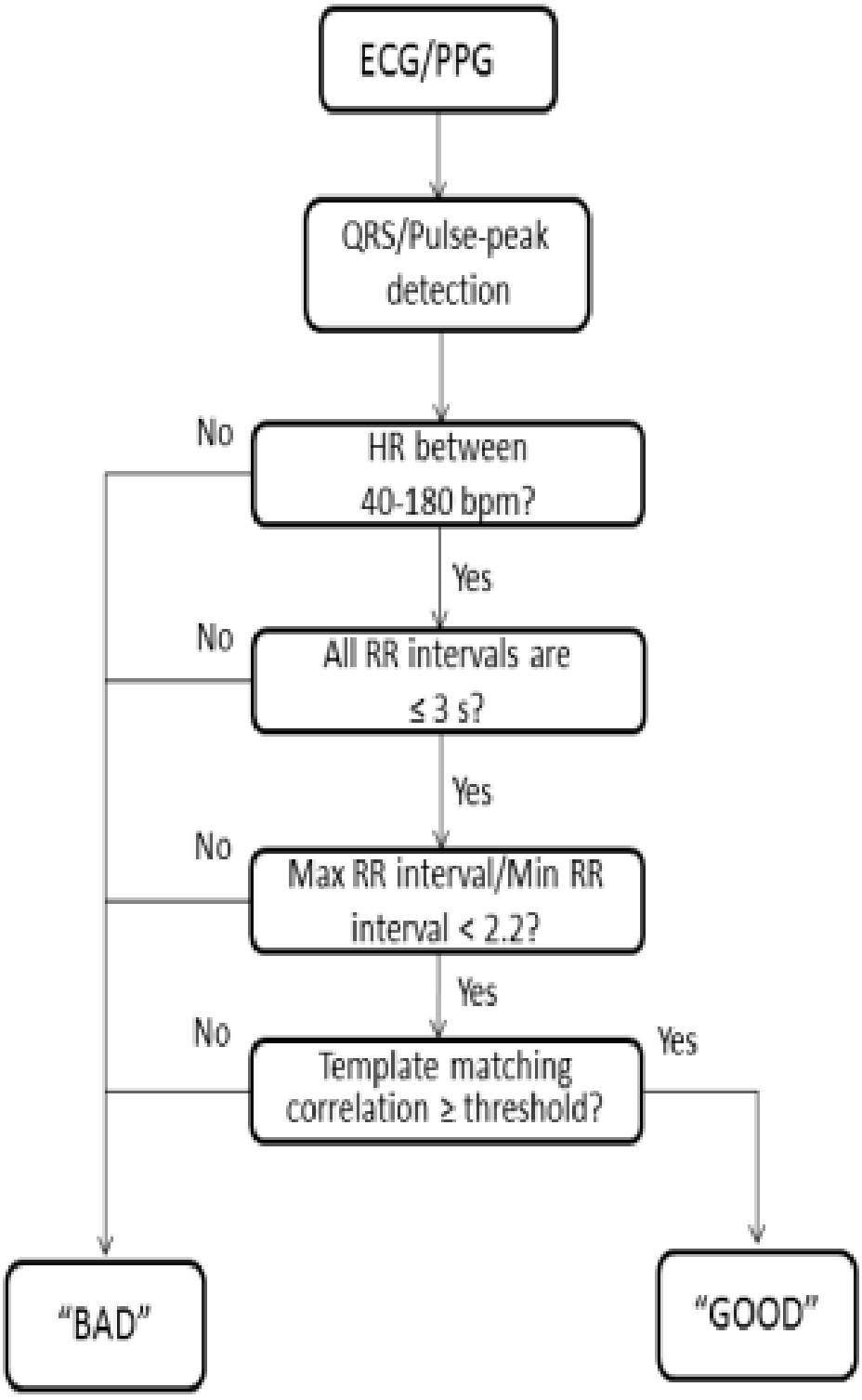



Figure 6.6: Flowchart of SQI algorithm *Source: [15]*

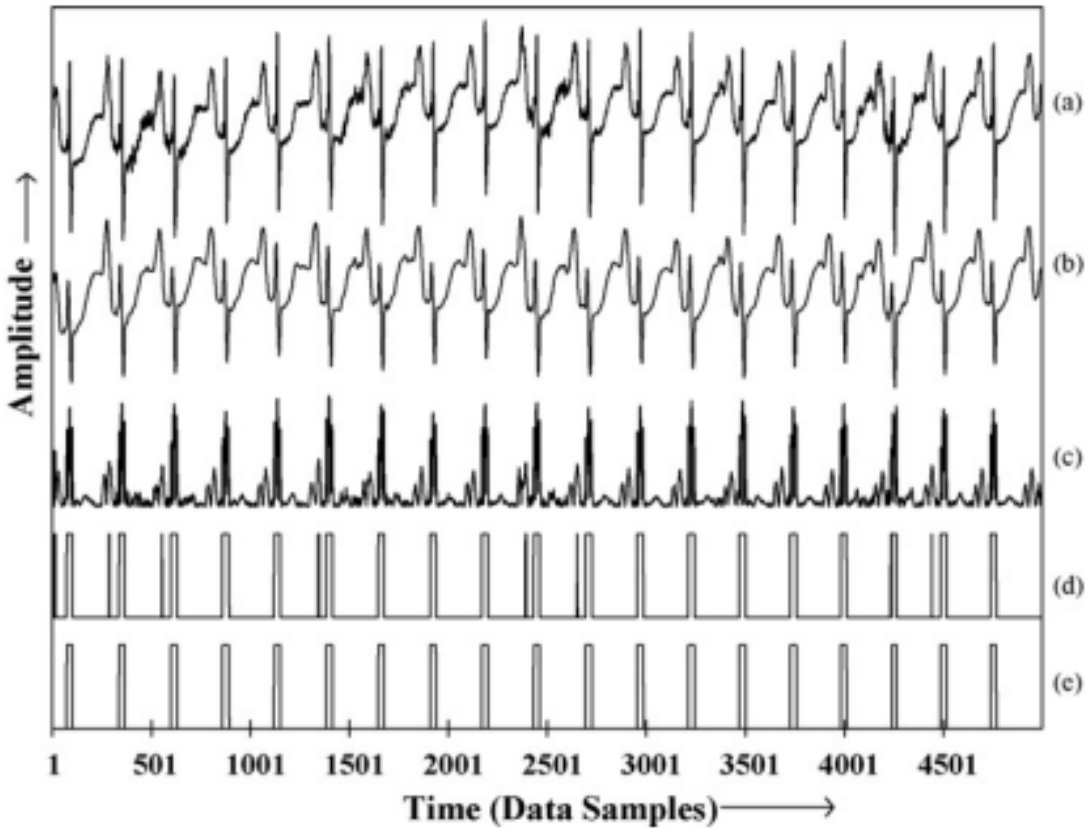



Figure 6.7: Plot d) K-means algorithm detecting QRS locations *Source: [34]*

# Chapter 7

# Summary and Conclusion

The goal of this project was to explore medical time-series data from patients in the ICU and to investigate whether models using ARIMA, cross-correlation and correlated windows could be beneficial in reducing false alarms in the ICU. Because hospitals suffer from a large number of false alarms, this is an important and relevant topic. In this project, we mentioned several insights regarding the behavior of specific medical signals and their effect on true or false alarms. From the PhysioNet Challenge 2015 Database, we were able to derive that signals, which caused a false alarm, were often noisier. This affected, that the ARIMA models performed on average worse when the alarms were false. Hence, we tried to use this effect to create a model that predicts, based on how well ARIMA was able to model the patients' signal, whether or not the signal caused a true or a false alarm to be raised. The results in section 4.3 showed that this approach was able to correctly classify a few patient records, but not enough to adopt the models in a real hospital environment. However, using the errors that ARIMA produces as a predictive model has the advantage that the model does not depend on the patient. Thus, it does not matter if one patient's signal differs greatly from another patient's signal. This can be a useful property because the normal state of a signal can vary among different patients.

The correlation-based models proposed in this project serve the purpose of trying out new approaches that evolve from cross-correlation. Therefore the objective was not to use any known models to achieve a very high score but to examine how cross-correlation of signals can identify changes in signals that could eventually reflect the health condition of a patient. The records were carefully analyzed and explored with plots and the changes that happen before the alarm is triggered can be recognized from the plots. The Single-column based model and the Between-column based model performed well but not in all alarms types. The models scored high in Tachycardia alarms where the signal is characterized by the high frequency of the QRS wave. Similar results are also received from the ARIMA models. However, the signal preprocessing methods used in ARIMA did not help to improve the correlation models. This could be because the sharper the signal the better it is for correlation, so cutting-off the signal frequency makes the signal smoother and reduce the amplitude values to fit other the nearest values. Another factor that could trick the models is the range of amplitude which is normal to shift from -1

to 4 mV. If the peak hits 2 or 4 mV that is not supposed to affect the alarms. The row-based model did not give good scores and has a high time complexity so it is not recommended to be used for real-time analysis.

The cross-correlation of windows model was established on the knowledge and insights that came from the work and efforts during the exploratory data analysis. While taking the window-based approach we were able to get local information about the values which also assisted in the later model parameter tuning such as window size which was changed multiple times. Afterward, the correlation between windows is used in the threshold and inside the model which is trained and tested on the same dataset as the previously mentioned models. The cross-correlation model performed slightly worse than the ARIMA and Correlation-based models which was also presented through the results.

Comparing the two types of alarms that were predicted by the model separately, Tachycardia and Bradycardia alarm types showed quite different results. The model performed better in Tachycardia alarms similarly to the previously discussed models but quite low on the Bradycardia. However, the results of both alarm types were slightly lower than the ones from the other models. In other words, the window correlation-based model was able to correctly predict some false alarms but the performance of the model was not to the level that was expected. Nevertheless, the work of modifying and using the synchronous windows, correlating them, and using that information for the model can be extended and potentially improve the classification of false alarms coming from this type of data.

In summary, this project reveals insights from our medical data exploratory with the main focus on alarms in the ICU. We suggest several models that could potentially reduce the number of false alarms, and provide results and observations that were gathered in the course of this project. Our results are a product of exploring medical datasets and trying out different models. However, more research is necessary to adopt any of the mentioned models.

# Appendix A

# MIMIC-III Database

The MIMIC-III Database was first considered because it contains a lot of medical data. However, it could not be used for creating models that could improve the false alarm rate in ICUs, because it did not contain well-labeled alarms. Hence, the Physionet Challenge 2015 database was used instead. However, a lot preliminary work was done on the MIMIC-III Database. It was helpful in the learning process, including how to read, plot and explore signal data, what the signals mean and how they correlate with each other, and finally, it provided data for the first steps in creating models. This chapter contains all the work that was done on the MIMIC-III Database.

## A.1 Database Overview

MIMIC-III is an openly available dataset developed by the MIT Lab. It contains health data from patients in the intensive care unit [24]. It consists of two parts, the clinical database and the waveform database. Both databases are linked with each other. Sections A.1.1 and A.1.2 will introduce these databases. Section A.2 will give an overview of the exploratory analysis that was done on both databases.

### A.1.1 MIMIC-III Clinical Database

The Clinical Database is a relational database and includes health-related data of patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. It is available from PhysioNet [21]. The database is useful for this project because it contains patient characteristics such as age, gender, diagnosis, prescriptions and mortality. It also contains charted data in the table CHARTEVENTS. The electronic chart displays patients' routine vital signs such as heart rate, pulse and respiratory rate. However, these vital signs are only recorded about once an hour and are therefore not ideal for data modeling. It contains a column which specifies for a given vital sign whether or not a warning was raised. Unfortunately, not a lot of information is provided on how those warnings were raised and whether or not they were real warnings or false warnings [24].

### A.1.2 MIMIC-III Waveform Database

The MIMIC-III Waveform Database contains signals from bedside patient monitors in intensive care units (ICU's). The signals in the waveform database vary depending on the choices made by the ICU staff. It can contain heart rate, respiratory rate, SpO2 and blood pressure. The time-series of vital signs are sampled once per second or once per minute [23]. There exists a matched subset of patients from both the clinical database and the waveform database. This subset is especially interesting because of the large amount of information available on the patients and because it is possible to merge the signals with the warnings of the Clinical Database. By analyzing the signals before the warnings and at times without any warnings can provide useful insights on how to reduce false alarms in the ICU.

## A.2 Exploratory Analysis

Most of the exploratory analysis for these databases were done on patient 85 because that patient is contained both in the clinical and in the waveform database and it contains several warnings.

The clinical database consists of 26 tables. The most interesting table for this analysis is the CHARTEVENTS table. It contains all the charted data available for a patient during their ICU stay [24].

| ROW_ID | SUBJECT_ID | HADM_ID | ICUSTAY_ID | ITEMID | CHARTTIME | STORETIME | CGID | VALUE | VALUENUM | VALUEUOM | WARNING | ERROR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2190 | 85 | 112077 | 291697.0 | 223751 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 160.0 | 160.0 | mmHg | 0.0 | 0.0 |
| 2191 | 85 | 112077 | 291697.0 | 223752 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 90.0 | 90.0 | mmHg | 0.0 | 0.0 |
| 2192 | 85 | 112077 | 291697.0 | 223769 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 100.0 | 100.0 | % | 0.0 | 0.0 |
| 2193 | 85 | 112077 | 291697.0 | 223770 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 92.0 | 92.0 | % | 0.0 | 0.0 |
| 2194 | 85 | 112077 | 291697.0 | 224161 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 35.0 | 35.0 | insp/min | 0.0 | 0.0 |
| 2195 | 85 | 112077 | 291697.0 | 224162 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 8.0 | 8.0 | insp/min | 0.0 | 0.0 |
| 2196 | 85 | 112077 | 291697.0 | 226253 | 2167-07-26 16:58:00 | 2167-07-26 18:59:00 | 20214.0 | 85.0 | 85.0 | % | 0.0 | 0.0 |
| 2197 | 85 | 112077 | 291697.0 | 220045 | 2167-07-26 17:00:00 | 2167-07-26 17:02:00 | 20214.0 | 100.0 | 100.0 | bpm | 0.0 | 0.0 |
| 2198 | 85 | 112077 | 291697.0 | 220179 | 2167-07-26 17:00:00 | 2167-07-26 17:02:00 | 20214.0 | 109.0 | 109.0 | mmHg | 0.0 | 0.0 |
| 2199 | 85 | 112077 | 291697.0 | 220180 | 2167-07-26 17:00:00 | 2167-07-26 17:02:00 | 20214.0 | 60.0 | 60.0 | mmHg | 0.0 | 0.0 |

Figure A.1: CHARTEVENTS Sample for patient 85

Figure A.1 shows an example of the CHARTEVENTS table for patient 85 (SUBJECT_ID). The SUBJECT_ID is useful because it can be used to merge different tables. Hence, it can be found that this patient has been admitted twice in an emergency, the first time due to aortic stenosis and 5 years later due to pneumonia. The ITEMID column contains a code which indicates the measurement type (e.g. heart rate) and VALUE contains the value of the measurement. The WARNINGS column is 1 if that specific row raised a warning and 0 otherwise.

The corresponding waveform database for patient 85 contains measurements of a lot of vital signs including heart rate, pulse, respiration and SpO2.

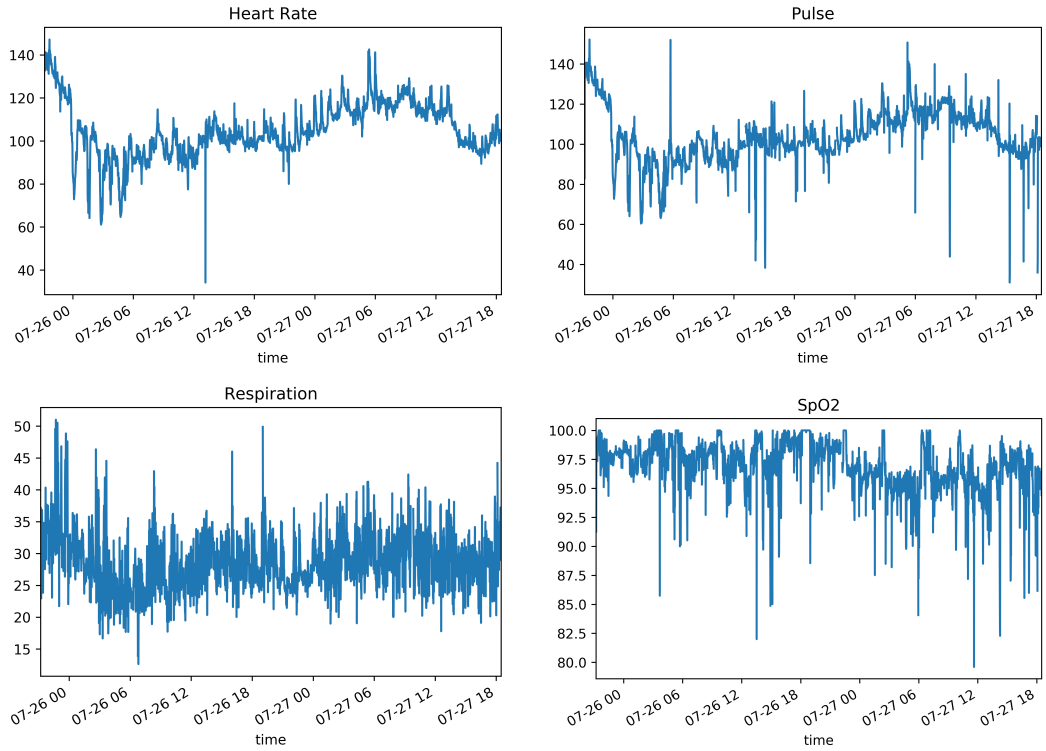



Figure A.2: Heart Rate, Pulse, Respiration and SpO2 signals for patient 85

Figure A.2 shows example plots of how these signals evolve over time for patient 85. The signals are sampled once per minute over a period of two days.

Figure A.3 shows the same signals with the warnings. The blue dots indicate when a warning was raised. Each warning comes with a label, e.g. Hemoglobin warning, which describes what kind of warning was raised. In figure A.3 all the warnings are considered, however, combinations of only a few specific warnings were also explored. From the plots it can be seen that where the heart rate, pulse and respiration decrease in the beginning, at least two warnings occurred at around the same time.

### A.2.1 Exploratory Analysis from Correlation Perspective

In the MIMIC-III database we have recordings of inpatients who spent their time in ICU. What makes this database specific is the fact that a lot of patients' records are available including demographic information. However, the alarms were not raised automatically by the machine but rather were identified manually by a nurse and labelled as 0 and 1 for no alarm and alarm respectively. Consequently if the nurse was not present then no alarms would be found. However, we know little about the criteria that were taken into account from the nurse to decide if an alarm should be raised. Eventually, the database was not a good fit to develop models on reducing the number of false alarms.

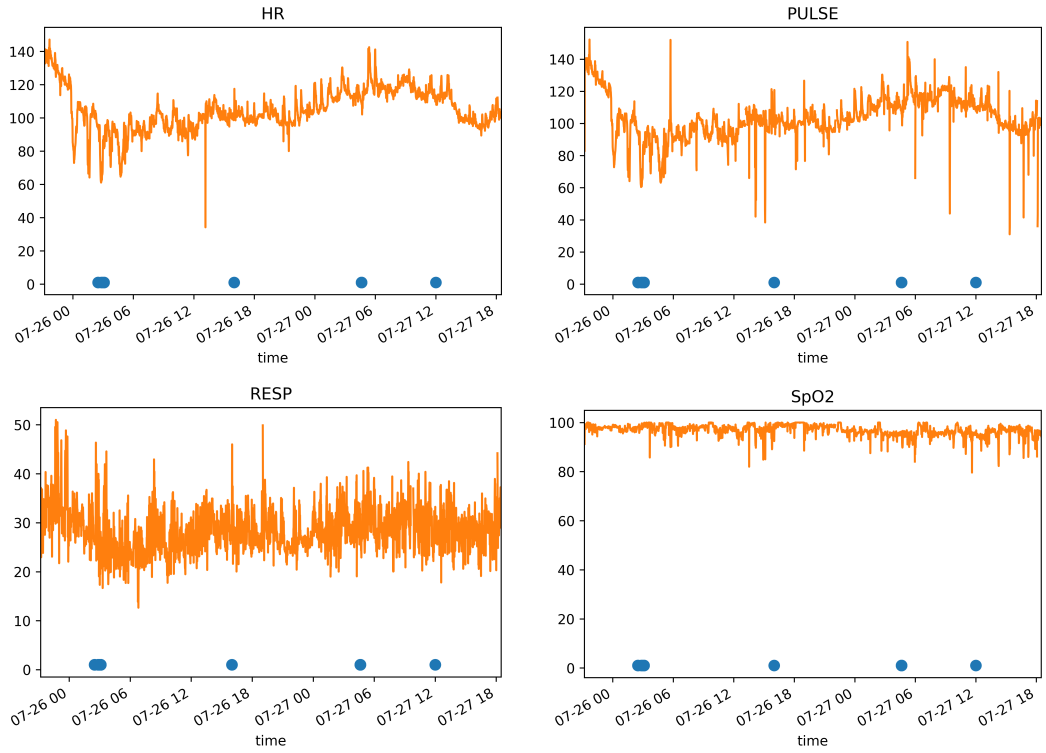



Figure A.3: Heart Rate, Pulse, Respiration and SpO2 signals for patient 85 with Warnings

The mimic-3 database is a relational database which is very well structured and can easily be imported in a database management system.

The equivalent database of the MIMIC-III, the waveform database came with a number of physiological signals few of which can be seen in Figure A.4.

| | Time | Hr | Pulse | NBPSys | NBPDias | NBPMean | Resp | SpO2 | SysDias |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 2167-07-25 21:11:31 | 133.966 | 0.000 | 0.0 | 0.0 | 0.0 | 31.333 | 0.000 | NaN |
| **2** | 2167-07-25 21:12:31 | 137.317 | 82.966 | 0.0 | 0.0 | 0.0 | 37.150 | 91.271 | NaN |
| **3** | 2167-07-25 21:13:31 | 141.167 | 140.250 | 0.0 | 0.0 | 0.0 | 34.866 | 97.400 | NaN |
| **4** | 2167-07-25 21:14:31 | 137.151 | 138.683 | 0.0 | 0.0 | 0.0 | 22.950 | 97.750 | NaN |
| **5** | 2167-07-25 21:15:31 | 132.767 | 134.299 | 0.0 | 0.0 | 0.0 | 35.400 | 99.217 | NaN |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2715** | 2167-07-27 18:25:31 | 102.650 | 101.583 | 0.0 | 0.0 | 0.0 | 37.233 | 95.967 | NaN |
| **2716** | 2167-07-27 18:26:31 | 101.451 | 100.334 | 0.0 | 0.0 | 0.0 | 35.333 | 95.667 | NaN |
| **2717** | 2167-07-27 18:27:31 | 100.966 | 100.634 | 0.0 | 0.0 | 0.0 | 32.783 | 95.450 | NaN |
| **2718** | 2167-07-27 18:28:31 | 101.483 | 99.683 | 0.0 | 0.0 | 0.0 | 32.484 | 95.833 | NaN |
| **2719** | 2167-07-27 18:29:31 | 103.267 | 102.866 | 96.0 | 73.0 | 77.0 | 29.000 | 94.417 | 1.0 |

Figure A.4: A number of signals present in the waveform database of the patient 85

In the waveform database we started finding out the correlation between signals. As the first step is to find out the overall correlation between signals, this help us understand which pairs of signals highly correlate throughout the entire record. In the Figure A.6 we can see how strong the signals correlate with each other. In the x axis we can see the signal names and in the y axis the Pearson correlation coefficient. When the signal is correlated with itself the correlation is very high and it reaches the peak. The most interesting part is when the correlation of a signal is high when it meets signals other than itself. In the case of patient with number 85 the Systole and Diastole are highly correlated, which is then followed by heart rate and pulse. Interestingly a number of signals correlate with Pulse which can be seen in the Figure A.6. At the x axis when signals meet Pulse the correlation is quite high with correlation coefficient of at least 0.2.

Based on the preliminary insights, the signals were further investigated. Since Systole and Diastole are highly correlated and this is also expected from the doctor's point of view [31] who say that if the difference or the so called Pulse pressure is greater than 40 a number of cardiovascular diseases can be predicted. For this purpose the signals can be seen separately in the Figure A.5.

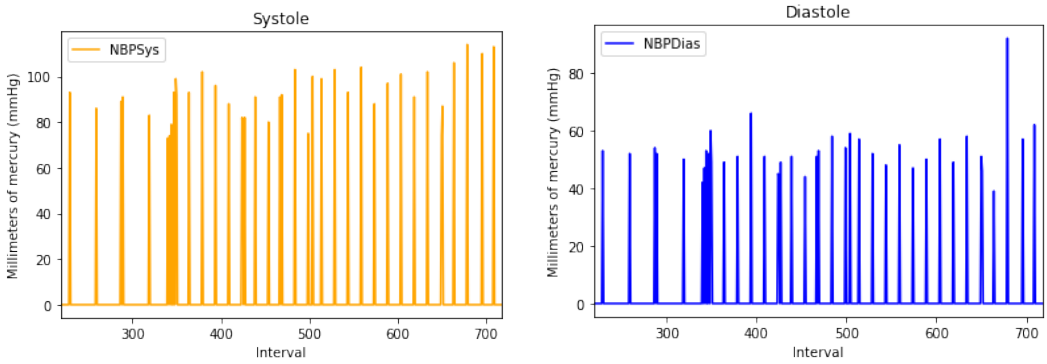



Figure A.5: Systole and Diastole

To find out how the correlation between Systole and Diastole evolves the sliding window technique was applied with window size 10 and 100 which can be seen in the Figure A.7. Towards the end of the record the correlation is lost where basically an alarm is raised.

Similarly we wanted to check how the level of oxygen or SpO2 is related with Respiration. The correlation seem to be volatile in the first plot with window size 10 but then when window size is 100 then similar pattern as in the Systole and Diastole correlation can be seen. This shows deterioration of the health condition.

The models applied in the Challenge 2015 do not necessarily apply to this dataset. Similar models were planned for MIMIC-III database but were not tested.

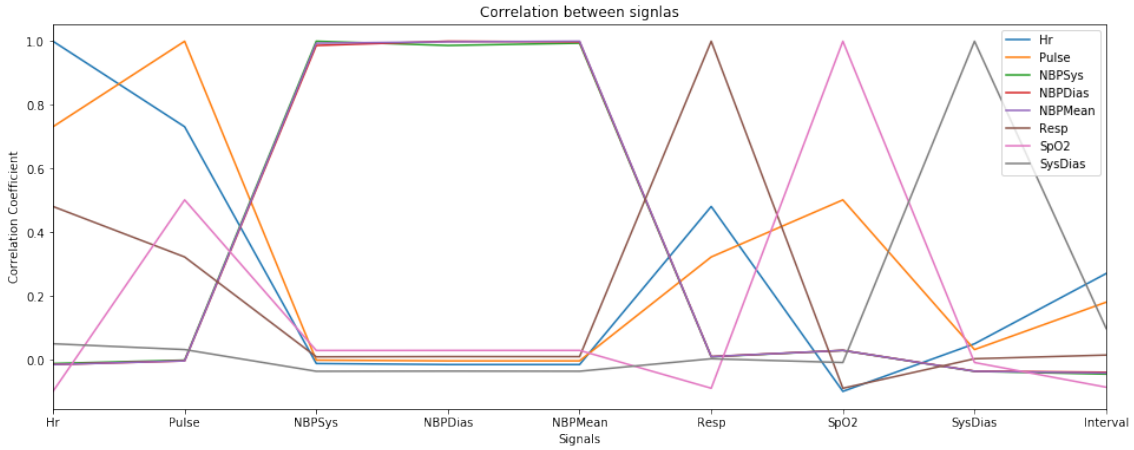



Figure A.6: Correlation between signals of the patient 85

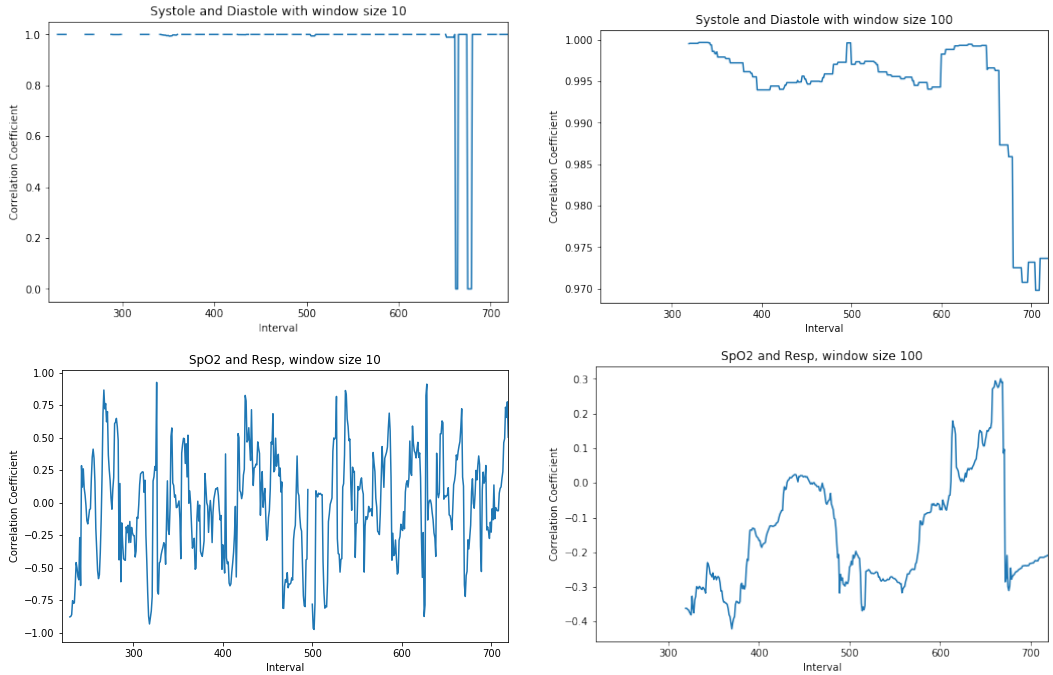



Figure A.7: Correlation of SpO2 with Resp and Systole with Diastole.

## A.3 Exploratory Data Analysis from Cross Correlation of Windows Perspective

Before using the Physionet Challenge 2015 dataset some time was spent learning about the medical metrics, how alarms were raised and discussing how we could approach the correlation of windows within the time series.

One of the first technical metrics that was explored and checked from the current dataset was correlation. Since in this part of our work the focus was on the correlation of fixed windows, the cross correlation of windows within the same signal was performed.

As we move on to the signal we will initially choose two windows of size 50 (rows) which we will explore and check the correlation of. These two windows come from different parts of the signal. The first window was taken within the first part of the signal and the second window from the last part out of the whole signal. Since these two windows are from very different stages of the signal we would expect low correlation due to the patients state also deteriorating towards the end of the signal before the alarm was raised.

Figure A.8 shows that the cross-correlation between the two windows is small. Moreover, the positions of the two lines of the HR signal, which represent the values in those windows, support this statement as well. Moreover, the signal was also

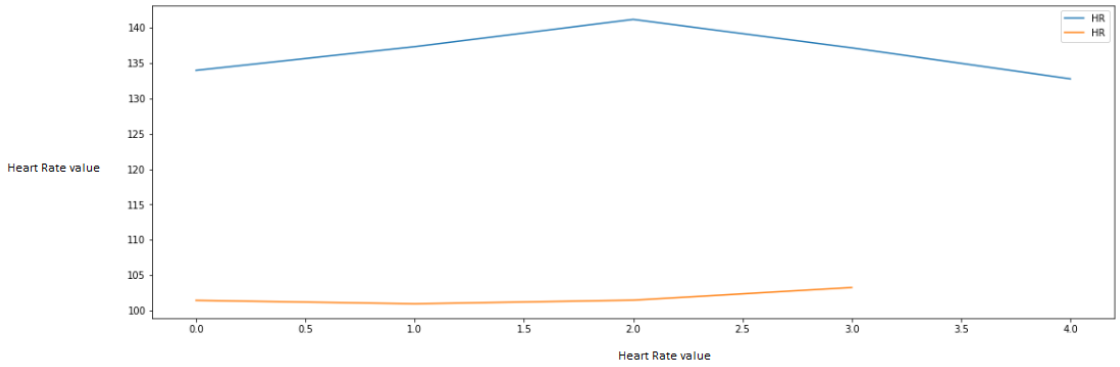



Figure A.8: Cross correlation of two windows within HR signal

checked for other components and metrics such as trend, residuals and seasonality. Initially, the seasonality of different metrics of a signal was visually checked and observed for patterns. Below on Figure the seasonality of the heart rate signal is visualized, we observe a sudden drop in the signal which is a possible indication of noise or sudden patient state deterioration.

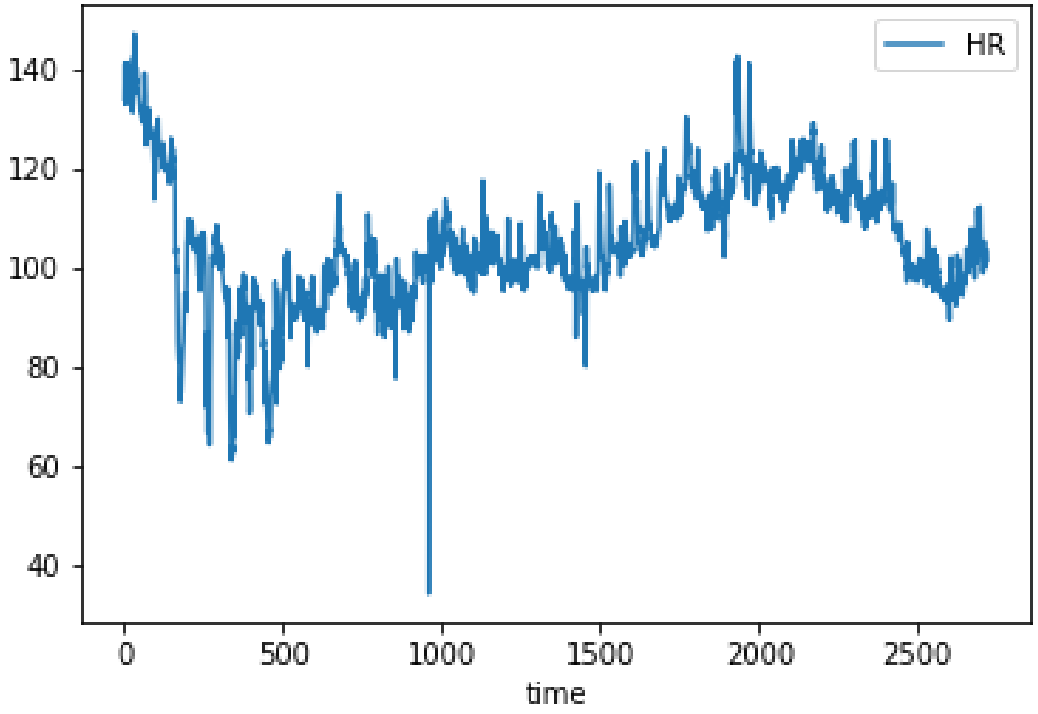



Figure A.9: Seasonality of reart rate signal

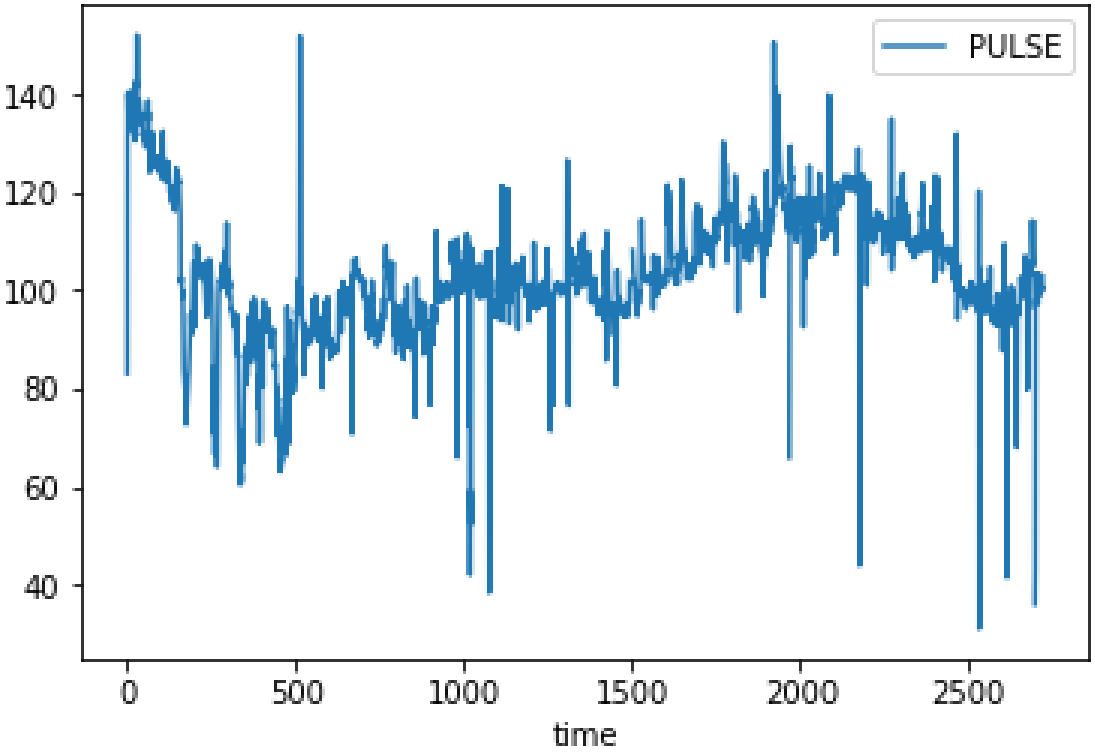



Figure A.10: Seasonality of pulse signal

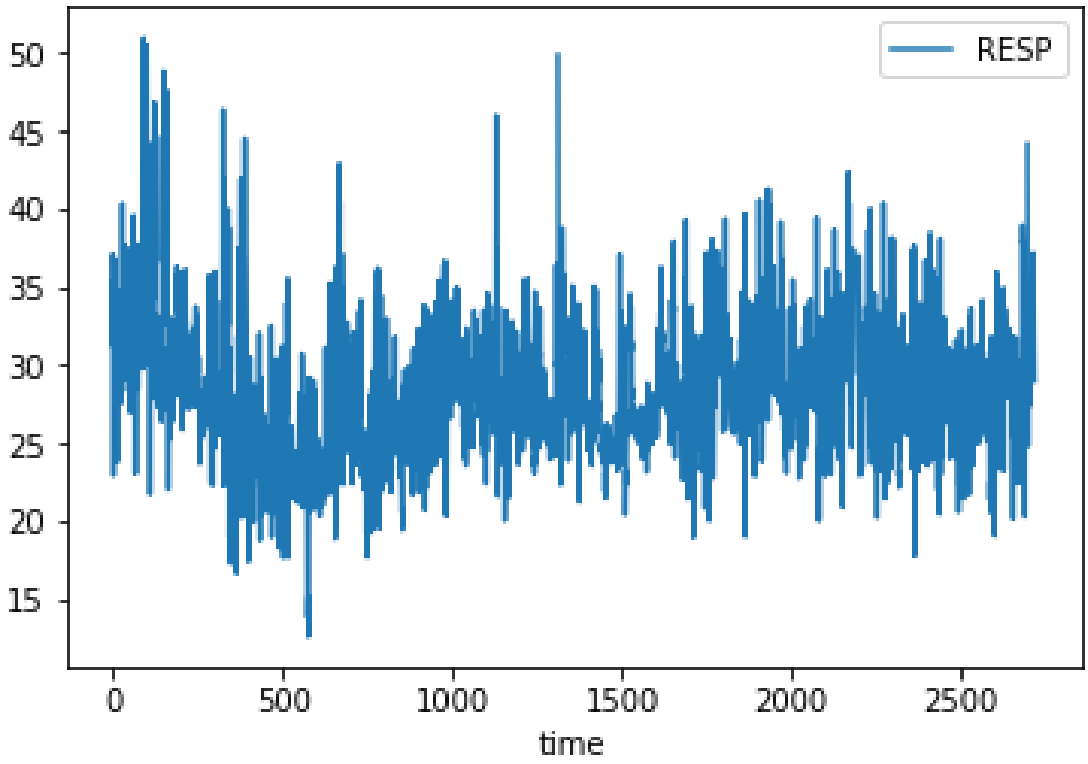



Figure A.11: Seasonality of respiratory signal

## A.4 ARIMA Model

The approach using ARIMA was first explored using the MIMIC-III database. As a first step, GridSearch for ARIMA(p,d,q) was run with the following possible parameters:

- $p \in \{0, 1, 2, 4, 6, 8, 10\}$
- $d \in \{0, 1, 2\}$
- $q \in \{0, 1, 2\}$

All possible combinations of parameters were tested for both one-step forecasts and multi-step forecasts with 20 steps. One-step forecasts only predict the value of the following time step. Multi-step forecasts with 20 steps predict the values for the next 20 time steps as rolling forecasts using already predicted values as input. This was done for the RESP signals and only for data between warnings, i.e. times where everything should be fine. GridSearch returned the optimal parameters (based on the mean squared error) as (6,0,0) for one-step forecast and (2,0,1) for 20-step forecast, which suggests that stationarity may not be an issue in this case because for $d > 0$ the model performed worse. Results also showed clearly that multi-step forecasting is not useful because the predictions gradually converge to the mean of the past signals.

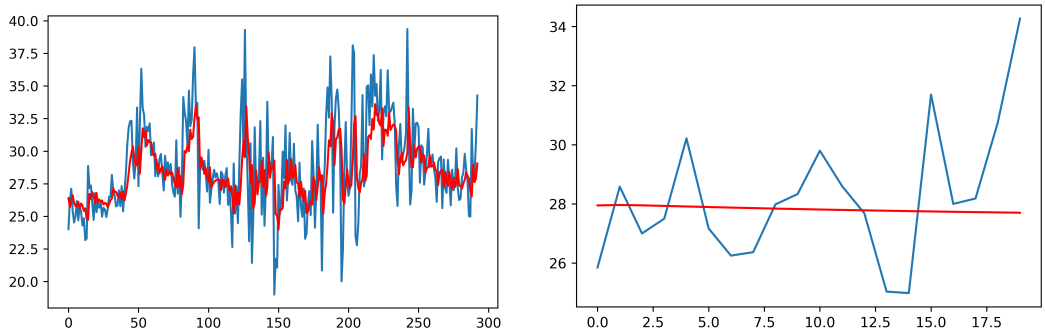



Figure A.12: ARIMA one-step forecast and 20-step forecast

Figures A.12 demonstrates this problem. On the left ARIMA(6,0,0) was run on a test set of RESP with one-step forecasting. The blue lines indicate the original values of RESP and the red lines show the ARIMA predictions. The red line can be seen as a smoothed version of the blue line and seems appropriate for modeling this data. On the right however, an ARIMA(2,0,1) 20-step forecast was used where the plot shows an example of a window with 20 time steps. The prediction does not model the actual values very well. Therefore, for future analysis only one-step forecasts was considered.

# Bibliography


[1] 12-lead ecg placement guide with illustrations. https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations. Accessed: 2019-03-26.

[2] Approach to ecg interpretation. https://www.healio.com/cardiology/learn-the-heart/ecg-review/ecg-interpretation-tutorial/approach-to-ecg-interpretation. Accessed: 2019-04-26.

[3] ARIMA Model – Complete Guide to Time Series Forecasting in Python. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Online; accessed 10-February-2020].

[4] Butterworth Filters. https://www.physik.uzh.ch/local/teaching/SPI301/LV-2015-Help/lvanlsconcepts.chm/lvac{_}butterworth{_}filters.html. [Online; accessed 5-May-2020].

[5] Determining rate. https://www.healio.com/cardiology/learn-the-heart/ecg-review/ecg-interpretation-tutorial/determining-heart-rate. Accessed: 2019-04-26.

[6] Overview of the peaks dectection algorithms available in python. https://pythonawesome.com/overview-of-the-peaks-dectection-algorithms-available-in-python/. Accessed: 2019-05-02.

[7] Qrs complex. https://en.wikipedia.org/wiki/QRS$_{c}omplex$. $Accessed$ : 2019 – 03 – 26.

[8] scipy.signal.butter. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.butter.html. [Online; accessed 28-May-2020].

[9] Time series models: AR, MA. https://medium.com/@sakiomb32309/ar-autoregressive-model-d65d5ce3c41. [Online; accessed 28-May-2020].

[10] *Basic EKG and rhythm interpretation Symposia*, January 2012.

[11] K. N. S. M. J. G. Behzad Ghazanfari, Fatemeh Afghah and J. Todd. An Unsupervised Feature Learning Approach to Reduce False Alarm Rate in ICUs. https://www.semanticscholar.org/paper/

An-Unsupervised-Feature-Learning-Approach-to-Reduce-GhazanfariAfghah/1eb00b289023402e7ebbe6227e1c0f1f8fa055c1, 2019.

[12] A. K. Bhoi, K. S. Sherpa, D. Phurailatpam, J. S. Tamang, and P. K. Giri. Multidimensional approaches for noise cancellation of ECG signal. *2015 International Conference on Communication and Signal Processing, ICCSP 2015*, (April):66–70, 2015.

[13] Q. Chaudhari. Sample Rate Conversion. https://wirelesspi.com/sample-rate-conversion/. [Online; accessed 20-April-2020].

[14] O. M. Cho, H. Kim, Y. W. Lee, and I. Cho. Clinical alarms in intensive care units: Perceived obstacles of alarm management and alarm fatigue in nurses. *Healthcare Informatics Research*, 22(1):46–53, 2016.

[15] P. C. D. C. D. V. Christina Orphanidou, Timothy Bonnici and L. Tarassenko. Signal Quality Indices for the Electrocardiogram and Photoplethysmogram: Derivation and Applications to Wireless Monitoring. https://www.researchgate.net/publication/264395822, 2014.

[16] G. D. Clifford, I. Silva, B. Moody, Q. Li, D. Kella, A. Shahin, T. Kooistra, D. Perry, and R. G. Mark. The physionet/computing in cardiology challenge 2015: Reducing false arrhythmia alarms in the icu. In *2015 Computing in Cardiology Conference (CinC)*, pages 273–276, 2015.

[17] G. D. Clifford, I. Silva, B. Moody, Q. Li, D. Kella, A. Shahin, T. Kooistra, D. Perry, and R. G. Mark. The PhysioNet/Computing in Cardiology Challenge 2015: Reducing false arrhythmia alarms in the ICU. *Computing in Cardiology*, 42:273–276, 2015.

[18] P. Couto, R. Ramalho, and R. Rodrigues. Suppression of False Arrhythmia Alarms Using ECG and Pulsatile Waveforms. Technical report, Faculdade de Ciencias e Tecnologia da Universidade Nova de Lisboa, Portugal, 2015.

[19] R. B. Ford and E. M. Mazzaferro. Section 4 - diagnostic and therapeutic procedures. In R. B. Ford and E. Mazzaferro, editors, *Kirk Bistner's Handbook of Veterinary Procedures and Emergency Treatment (Ninth Edition)*, pages 442 – 550. W.B. Saunders, Saint Louis, ninth edition edition, 2012.

[20] B. Ghazanfari, F. Afghah, K. Najarian, S. Mousavi, J. Gryak, and J. Todd. An unsupervised feature learning approach to reduce false alarm rate in icus. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 349–353, 2019.

[21] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[22] A. L. Goldberger, Z. D. Goldberger, and A. Shvilkin. Chapter 3 - how to make basic ecg measurements. In A. L. Goldberger, Z. D. Goldberger, and A. Shvilkin, editors, *Goldberger's Clinical Electrocardiography (Ninth Edition)*, pages 11 – 20. Elsevier, ninth edition edition, 2018.

[23] A. Johnson, T. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, and M. RG. Mimic-iii, a freely accessible critical care database. http://dx.doi.org/10.1038/sdata.2016.35, 2016.

[24] A. E. Johnson, T. J. Pollard, L. Shen, L. W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:1–9, 2016.

[25] K. Kleber. 5-lead ecg interpretation, electrocardiogram tips for nurses. https://www.freshrn.com/5-lead-ecg/. Accessed: 2020-03-26.

[26] MathWorks. medfilt1. https://ch.mathworks.com/help/signal/ref/medfilt1.html. [Online; accessed 28-May-2020].

[27] S. Meek and F. Morris. Abc of clinical electrocardiography: Introduction. i—leads, rate, rhythm, and cardiac axis. *BMJ (Clinical research ed.)*, 324:415–8, 03 2002.

[28] I. Memorang. Cardiac ECG. https://www.memorangapp.com/flashcards/184208/Cardiac+ECGs/. [Online; accessed 25-May-2020].

[29] S. Palachy. Stationarity in time series analysis. https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322. [Online; accessed 10-February-2020].

[30] F. Plesinger, P. Klimes, J. Halamek, and P. Jurak. False alarms in intensive care unit monitors: Detection of life-threatening arrhythmias using elementary algebra, descriptive statistics and fuzzy logic. *Computing in Cardiology*, 42:281–284, 2015.

[31] S. G. Sheps. Pulse pressure: An indicator of heart health? https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/expert-answers/pulse-pressure/faq-20058189. Accessed: 2020-10-26.

[32] P. Shirkey. Butterworth Filters in C#. https://www.centerspace.net/butterworth-filter-csharp, 2008. [Online; accessed 28-May-2020].

[33] S. O. Soma Halder. Stationarity of a time series models. https://www.oreilly.com/library/view/hands-on-machine-learning/9781788992282/15c9cc40-bea2-4b75-902f-2e9739fec4ae.xhtml. [Online; accessed 28-May-2020].

[34] N. L. S.S. Mehta, D.A. Shete and V. Chouhan. K-means algorithm for the detection and delineation of QRS-complexes in Electrocardiogram. https://www.researchgate.net/publication/246279158_K-means_

algorithm_for_the_detection_and_delineation_of_QRS-complexes_in_Electrocardiogram, 2009.

[35] H. Victoria. Bipolar and unipolar ecg. what's the difference? https://codeoneapp.com/2018/07/11/unipolar-bipolar-ecg/. Accessed: 2019-04-26.

[36] K. Vigneshand and T. Lakshman. Enhancing Accuracy of Arrhythmia Classification by Combining Logical and Machine Learning Techniques. Technical report, The University of Texas at Dallas, Department of Electrical Engineering, 2015.

[37] E. M. I. Wan-Tai M. Au-Yeung, Ashish K. Sahani and A. A. Armoundas. Reduction of false alarms in the intensive care unit using an optimized machine learning based approach. https://doi.org/10.1038/s41746-019-0160-7, 2019.