1 BACKGROUND AND RELATED WORK

This section provides background information on time series and ECGs, as well as methods to analyze them.

TODO make this a more complete introduction, give a verbal table of contents with reference to the different sections

1.1 Time Series and Time Series Analysis

-series-intro

This subsection will provide background information on time series and time series analysis methods. A time series is a set of values recorded at specific times. A common form of time series are discrete-time time series (often simply called discrete time series). Discrete time series are time series whose values are recorded at discrete points in time, the most common example of this are time series with values recorded at fixed intervals. Continuous-time time series are time series that are recorded continuously over a certain interval brockwell2016 [1]. Time series that contain a single value for each moment in time are called univariate time series, while time series that record multiple values at each moment in time are called multivariate time series anacleto2020 [2]. Time series are used in many disciplines to record information on time-dependent processes, e.g. stock prices in economics, the sun's activity in physics, or the heart's activity in medicine. Time series can be recorded digitally, physically, or, if they were recorded physically, can later be digitized. The recorded data can then be used to gain insight into the processes that were studied. To gain insight using a time series, the relevant information needs to be extracted from it–a process that is often called data mining. Data mining of time series is a vast discipline that, among others, includes lin2003 aghabozorgi2015 [3, 4]:

- visualization (graphical representation),
- forecasting (predicting future behavior),
- indexing (finding the most similar time series to a given one),
- clustering (dividing time series into groups of similar ones),
- anomaly detection (detecting parts that are not "normal" or do not fit certain parameters),
- classification (assigning a label based on its features, e.g. "sick" and "not sick"), and
- summarization (reducing the complexity–often length–while preserving important features).

Challenges for time series analysis include the often very large datasets that are difficult for humans to analyze and take up considerable digital storage space. Analyzing very large datasets requires a large amount of computational power because most data mining algorithms become less efficient with larger datasets lin2003 [3]. To mitigate this issue, time series dimension reduction (also known as dimensionality reduction or time series representation) is used. Dimension reduction transforms a "raw" (unmodified) time series into a representation that is simpler but nonetheless resembles the raw time series. This can be achieved by either using a method that reduces the number of values in a time series, or by extracting only the relevant features from the time series. According to aghabozorg shieh2008 [4, 5], there are four types of dimension reduction methods:

1. data dictated,
2. non-data adaptive,

dr-methods:01

3. model-based, and
4. data adaptive.

Methods 2–4 have their dimension reduction factors set by user-defined parameters. This means that the user can determine how much the dimension of the data should be reduced [4].

1.1.1 Data dictated representation

Data dictated methods derive their compression ratios from the data automatically, the most common form of this method is the clipped representation [4]. This representation simply transforms the raw time series into a sequence of 1s and 0s. A data points is assigned a 1 if its values is larger than the mean value of the time series, and a 0 otherwise. A sequence of 1s and 0s can be further compressed using various methods from computer science, finally yielding a very large compression ratio of 1057:1 [6].

1.1.2 Non-data adaptive representation

Non-data adaptive methods operate on time series segments with a fixed size to reduce the dimension and they are useful for comparing multiple time series with each other. These methods include the Discrete Wavelet Transform (DWT), the Discrete Fourier Transform (DFT), and the Piecewise Aggregate Approximation (PAA) [4]. The DWT uses wavelets, a limited-duration wave with an average value of 0, which represents both time and frequency information. The DWT is calculated using a series of filters applied to the signal. In [7], the DWT is used to detect beats in ECG signals and achieves a 0.221% detection error rate. The Fast Fourier Transform, an optimized form of the DFT, decomposes the its input signal into many sinus waves of different frequencies. In [8] it is used in conjunction with a machine learning model to achieve a beat classification accuracy of 98.7%. The PAA is part of the process of the SAX representation, thus it will be covered in **TODO** refer to the appropriate methods section

1.1.3 Model-based representation

Model based methods use stochastic methods such as Hidden Markov Models (HMM) and the Auto-Regressive Moving Average (ARMA) [4]. A HMM was used in [9] to cluster electroencephalograph recordings (measuring the brain's electrical activity). It was found that their methods was competitive with other established methods in classifying electroencephalograph signals. An auto-regressive model can be used to correctly identify a specific type of arrhythmia in an ECG and to group the occurrences of this arrhythmia together [10].

1.1.4 Data adaptive representation

Data adaptive methods use non-fixed size segments and aim to fit the raw data most closely. Examples of data adaptive methods are the Piecewise Polynomial Approximation (PPA), Piecewise Linear Approximation (PLA), Piecewise Constant Approximation (PCA), and SAX [4]. PPA can be used to compress and ECG by approximating it using polynomials. With second-order polynomials, ECGs can be compressed with a minimal level of distortion [11]. The authors of [12] use a modified PLA representation with adaptive ECG segmentation to successfully reconstruct the 12 standard leads of an ECG from only 3 leads. Using adaptive PCA as the dimension reduction method, the preprocessing and segmentation of ECGs can be significantly sped up while maintaining accuracy comparable to precious methods [13]. The SAX representation will be covered in detail in TODO refer to the SAX section and the following subsection 1.1.5 will provide background on the method and its variations.

1.1.5 SAX representation background

A particular dimension reduction method is SAX. Introduced by Lin, Keogh, Lonardi, and Chiu, SAX is a symbolic time series representation method for univariate time series. The authors felt that the symbolic methods available in 2003 did not provide the desired dimension reduction, did not correspond to the raw data accurately enough, and could not be applied to a subset of the total data. SAX uses the averaging of a user-defined number of segments and the labeling of segments with letters to reduce the dimension of the time series data. The number of letters, called the alphabet size, can also be chosen by the user and influences the dimension reduction. The distance between two time series in the SAX representation is guaranteed to resemble the distance between the two raw time series, this is called the distance measure. Since its creation, SAX has found widespread use in data mining and many researchers have attempted to modify and improve it.

The SAX distance measure has been improved to include the standard deviation [14] and a measure of the trend of each averaged segment [15, 16]. Extended SAX modifies SAX to include the minimum and maximum values of each segment for improved representation of the raw data [17] while 1d-SAX incorporates a linear regression over each segment into SAX [18]. A combination of SAX and a polynomial approximation was used to speed up the SAX method [19]. To improve the indexing performance of SAX, iSAX introduced convertible alphabet sizes, allowing SAX representations with different alphabet sizes to compared with each other and indexed into a tree structure [5]. iSAX 2.0 improves the iSAX index by reducing its computational complexity, enabling it to index a time series that has one billion elements, something that SAX or iSAX cannot do [20]. To perform time series anomaly detection using SAX, Keogh, Lin, and Fu introduced Heuristically Ordered Time series using SAX (HOT SAX) in 2005. Specifically, the authors attempt to detect time series "discords", a subsequence of a time series that is most different from other segments of the time series. This can theoretically be done by simply comparing all subsequences of the raw time series to all other segments, but this approach is not feasible for long time series because of its complexity. Thus, HOT SAX utilizes SAX to reduce the dimensionality and complexity of the time series and then sorts the resulting SAX segments to speed up the discord detection. The authors suggest further research to investigate the use HOT SAX on multivariate time series [21]. For an in-depth description of this method, please refer to TODO refer to the methodology section of HOT SAX.

TODO mention this in the methodology section as supporting information

SAX and its variants have also been used for the analysis of multivariate time series. SAX-

ARM combines the SAX representation with association rule mining (identifying rules and implications found in the data, i.e. parameter a influences parameter b) to analyze multivariate time series and discover the rules underlying the data [22]. Anacleto, Vinga, and Carvalho introduced MSAX in 2020 and thus expanded the use of SAX to multivariate time series. They utilize multivariate normalization with a covariance matrix and a modified distance measure to achieve this. To analyze their method, the authors use MSAX and SAX in a classification task based on multiple multivariate time series datasets. For these multivariate datasets, SAX was applied to each of their individual time series and those results were combined. Their analysis found that, overall, SAX applied in this way is superior to MSAX when it comes to classification accuracy. In 6 of the 14 tested datasets, SAX was significantly more accurate, in 2 of the MSAX was more accuarate, and in the remaining 6 their performance was not significantly different. It should be noted that in the ECG dataset they tested, the accuracy of SAX (~87%) was slightly higher than that of MSAX (~84%), but not significantly so. Anacleto, Vinga, and Carvalho suggest that in future research MSAX should be applied to electronic health records (e.g. ECGs) and that it should be applied to other time series data mining applications besides classification [2]. MSAX will be thoroughly presented in **TODO** refer to methods section. **TODO** mention this in the methodology section as supporting information Another application of SAX to multivariate data used it to visualize multivariate medical test results and enable their analysis [23]. Resource-aware SAX is a SAX variant developed to analyze ECG using a mobile device like a mobile phone. The method takes advantage of the computational efficiency of SAX to perform the ECG analysis on the device and even preserve its battery life. Another application of the SAX method to ECGs is [24], which uses SAX with an added binary measure of the trend of each segment to detect ECG anomalies, achieving a recall value of 98%. The section 1.2 below will elaborate on ECGs and methods of their analysis.

1.2 ECGs and ECG Analysis

The following subsection covers the ECG and methods used in its analysis. Luigi Galvani noted the electrical activity in muscles 1786, but the history of the ECG only started in 1842, when Carlo Matteucci showed the electrical activity of a frog's heartbeat. In the 1870s, it was discovered that each heartbeat is characterized by electrical changes. Then, in 1901-1902, Willem Einthoven created the first ECG recording of a human heartbeat using using 3 leads connected to the limbs of the patient. Einthoven was the first to publish an ECG waveform with the now standard annotations P, Q, R, S, and T for the different features (see Figure 1.1). He would receive the 1924 Nobel Prize in medicine for his invention of the electrocardiograph. As a result of further development, the 12-lead ECG that we know today was created [25, 26]. The 12-lead ECG is comprised of 6 chest leads (measurements of electrodes on the chest) numbered consecutively V1 to V6, as well as 6 limb leads (measurements of electrodes on the limbs) called I, II, III, aVR, aVL, aVF [27].

1.2.1 What is an ECG?

An ECG records the electrical activity that accompanies the contraction and relaxation of the heart muscle. The sinuatrial node, which can spontaneously give off an electrical pulse, initiates the heart

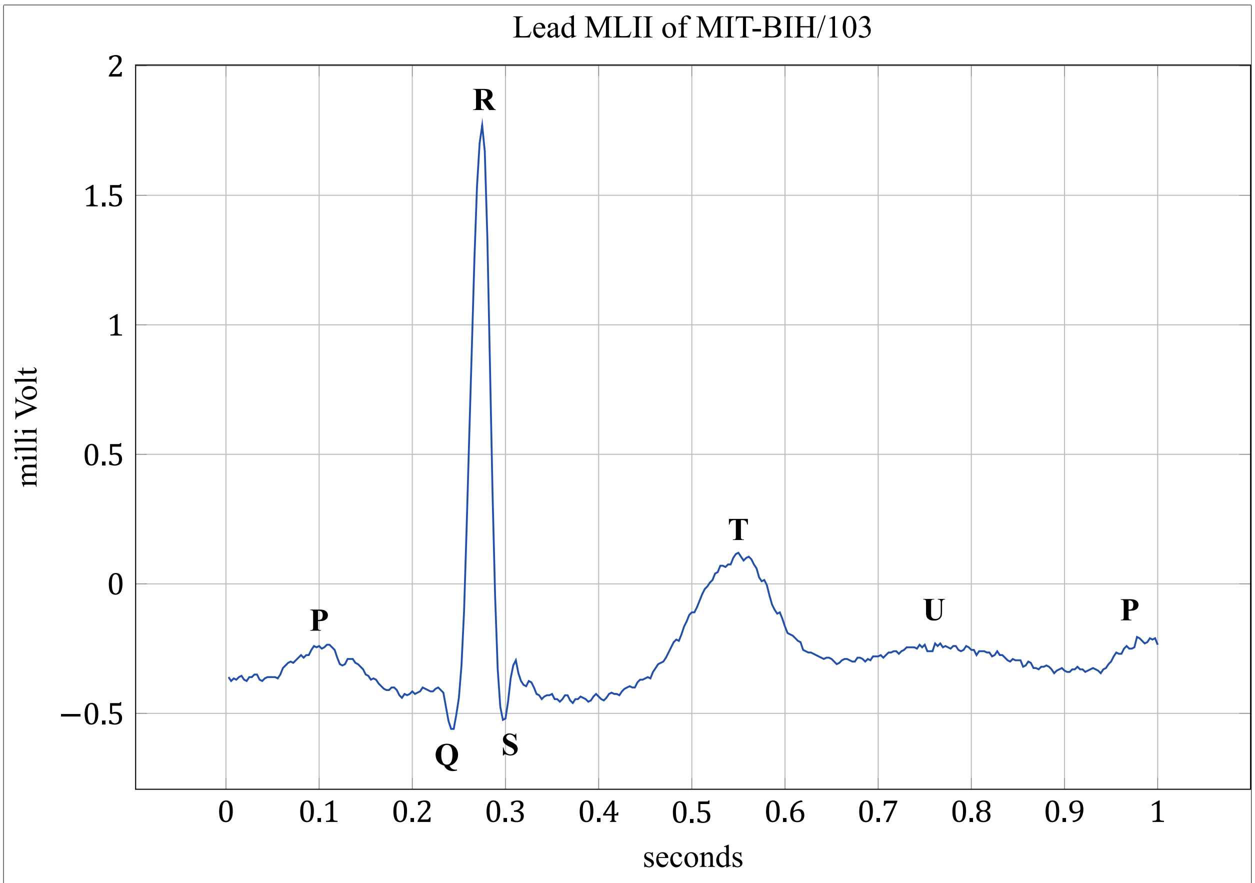

Figure 1.1: Annotated ECG of one heartbeat. This graph is based on lead II, data points 2031–2390 of recording 103 of the MIT-BIH database [29, 30].

beat. Its pulse is conducted through the heart by other specialized fibers, causing the heart to beat. The conduction of electricity is facilitated by Sodium, Calcium, and Potassium ions flowing in and out of cardiac cells [28]. Figure 1.1 shows a ECG wave of a single heartbeat from record 103 of the MIT-BIH database [29, 30] (for more information on the database, see section 1.2.3). The P wave is caused by the depolarization of the atrial node, which allows blood to flow into the heart. The QRS complex, as it is called, is the result of ventricular depolarization and represents the action of pumping blood out of the heart. The T wave is caused by ventricular repolarization in preparation for the next heartbeat. The U wave, only present in about 25% of people, is thought to he caused by mechanical-electric feedback [28, 31]. The last P wave is part of the next heartbeat, which is not shown in Figure 1.1.

The waves and complexes shown in Figure 1.1 are the object of ECG analysis. Changes in their shape, duration, or height can indicate heart conditions. Becker list some of the features relevant for ECG analysis [28]:

- The regularity of the rhythm: are the intervals between the QRS complexes and P waves regular?
- The shape of the QRS complex: do they have similar shape and duration?
- The regularity of the P waves: are the P waves similar and is the interval between P wave and QRS complex similar?
- Is the heart rate regular: measuring the time between QRS complexes can be used to calculate

the heart rate, is this heart rate in the normal range?

- Do the waves and complexes come in the same order each time: each cycle should consist of a P wave, QRS complex, T wave.

Using an ECG to diagnose a cardiac condition is difficult in practice. Small changes in the components of the ECG can be indicators of diseases and those changes can be overlooked, even by trained and specialized physicians. The chance to make a mistake is even higher for non-specialized physicians and trainees [25, 32]. The American Heart Association (AHA) estimates that a physician needs to read at least 500 ECGs with the help of an expert before becoming proficient. One reason for this is that the number of diagnosis that can be performed using an ECG is vast. The AHA lists 88 different conditions and an additional 22 diagnoses related to diseases and conditions that may not directly affect the heart, such as hypothermia or tremors caused by Parkinson's disease [33].

Two types of heart conditions that an ECG can detect are cardiac arrhythmias and ischaemic heart disease. Cardiac arrhythmia is a variation of the heart rate or rhythm that does not have a reasonable cause. In other words, heart rate or rhythm variations caused by physical activity could not be considered arrhythmias, while significant variations in a resting state may [34]. In an ECG, arrhythmia is most apparent in changes in the interval between the QRS complexes. Ischaemic heart disease is the main cause of death world-wide [35]. Ischemic heart disease is characterized by restricted blood flow to an area of the heart, causing it to not receive enough blood and oxygen. Blood flow restriction is caused by a blockage (or narrowing) in a blood vessel supplying the heart muscle. An artery can be blocked by a blood clot, but the most common cause is plaque buildup, called atherosclerosis. If the circulation to the heart is completely blocked, the cells in the heart muscle begin to die. This is called myocardial infarction, more commonly known as a heart attack. The deprivation of oxygen the heart experiences leads to the characteristic chest pain commonly associated with heart attacks [36]. In an ECG, ischaemic heart disease can be diagnosed based on changes in the ST segment and the T wave. The diagnosis of ischaemic heart disease and other heart diseases is time sensitive. If a patient has suffers from a heart attack, treatment has to be started as soon as possible. Some forms of treatment are most effective in the first 3 hours after symptom onset and lose most of their effectiveness after 9 to 12 hours. The diagnosis required for treatment to begin should thus be as quick as possible. The real-time information delivery of an ECG is an advantage in this situation, even though there are more time-consuming methods that can deliver more accurate results than an ECG [37].

1.2.2 Computerized ECG analysis

The widespread use of ECGs and the time-sensitive nature of their application as diagnostic tools makes errors, delays, or inconsistencies in their interpretation unacceptable. A recent approach to minimizing this problem is the application of computer technology in ECG recording, storage, and analysis [38]. Time series analysis methods can also be applied to ECGs because ECGs simply represent discrete multivariate time series. As discussed in section , multivariate time series are time series that contain more than one value at each point in time, while discrete time series are time series that are measured at discrete points in time or at set intervals. ECGs fulfill both of these requirements, as all modern ECGs contain at least 2 leads, most of them 12, and they have set sampling rates, given in samples per second. The common steps of computerized ECG analysis, following [38], are:

omp-ecg-an:01 1. signal acquisition and filtering,

omp-ecg-an:02 2. data transformation,

omp-ecg-an:03 3. waveform recognition,

omp-ecg-an:04 4. feature extraction, and

omp-ecg-an:05 5. classification or diagnosis.

Step 1 comprises the digital recording of ECG signals or the digitizing of paper-based ECG records. For either processes, the AHA recommends a sampling frequency of 500 samples per second. ECG filtering is performed to remove noise introduced by patient movements, power line interference, and other factors [38]. This filtering, or denoising, is often performed using digital filters. Their drawbacks are that they only filter out very specific frequencies. Because noisy ECGs contain different types of contaminations, digital filters can be inaccurate. Using wavelet transforms for denoising has the advantage that noise can be more precisely targeted and the clean signal reconstructed afterwards. Choosing appropriate wavelet parameters can be challenging, but methods to optimize this process have been proposed [32]. Step 2 uses the same types of methods for dimension reduction that were discussed in sections 1.1.1–1.1.4 for time series and shall not be repeated here. This includes SAX, which has been successfully applied to ECG analysis [24]. MSAX has, at the time of writing, to the author's knowledge not been applied to ECG analysis. HOT SAX has been used in [39] to detect anomalies in ECGs. It was found to detect anomalies, but it exhibited a larger amount of false identifications than competing methods.

Steps 3 and 4, the waveform recognition and feature extraction steps, are signified by extracting features that are relevant for diagnosis from the many points of the ECG. This process can also be aided by an appropriate representation chosen in the previous step. The main features targeted for extraction are the PQRST features shown previously in Figure 1.1. The Fast Fourier Transform (see section 1.1.2) provides a way of analysing the frequency domain of the ECG signal, enabling the detection of the QRS complex [8] and other features [40]. The missing time information in the Fast Fourier Transform can lead to difficulties in detecting time-dependent features. The short-time Fourier Transform adds time information to the fast Fourier Transform's data. This can increase the accuracy of the feature extraction. It has the drawback in the tradeoff between the time and frequency resolutions. Wavelet transforms can also be used for feature extraction [7]. They have the advantage that they are suitable for all frequency ranges. Choosing the right wavelet base for the desired application can be a challenge. The discrete wavelet transform is the most widely used wavelet transform, thanks to its computational efficiency. Statistical methods are also used to extract features from ECGs; those methods are generally less affected by noise in the signal [32].

After the features of the ECG have been extracted, it is often necessary to further reduce the number of features. The reason for this is that a large number of features, despite the high accuracy their analysis may yield, require a high amount of computation to classify. This lengthy computation can negate the advantages gained by high accuracy. Feature reduction sacrifices a certain amount of information and sometimes precision, but significantly speeds up the classification. There are two approaches to achieve this. First is feature selection, a process that attempts to select a subset of the original data that adequately describes the whole data. Feature selection can be performed by a filter that filters out unnecessary attributes based on some metric. This method is relatively simple, but the filtering process removes data and thus negatively impacts the precision of further steps. The second method, feature extraction, on the other, hand uses dimension reduction methods to keep as much of the original information as possible. Principal component analysis preserves as much of the variance in the original data as it can [7]. Other algorithms focus on separating classes of data, pattern recognition, or retaining the structure of the original data [32]. Here, again, the time series representation methods discussed in sections 1.1.1–1.1.4 can be applied.

Finally, the extracted features can be classified; this is stage 5. In this stage judgements are made based on the prepared input data and the result should be a disease diagnosis. Traditionally, this process is performed by a trained professional, as discussed in section 1.2.1. In the early stages of computerized ECG analysis, classification was performed by algorithms based on human actions when reading an ECG. Those algorithms were basic and not particularly accurate. Currently, the classification at the end of the preparation process is performed by a machine learning algorithm. Such models include the k-nearest-neighbors model which classifies points into groups but which is very expensive to calculate for high-dimensional data. Support vector machines are used for pattern recognition and are able to work with small samples. Artificial neural networks are robust and can work with complex problems, they are generally more accurate than support vector machines [8]. The newest approach is to forego the stages discussed here and use a single neural network to perform all the required tasks “end-to-end”. These networks are fed raw data perform steps 1.1.1–?? internally, as a single model [32]. This approach is relatively new and still actively researched. The previous approach, too, is enjoying active research attention.

1.2.3 ECG databases

07:ecg-db

A very important element of computarized ECG analysis is the training data. This data is used to train algorithms like neural networks, to manually tweak parameters of methods like SAX, or to validate and test prepared models. To fulfill these criteria, the data must be freely available to other researchers to replicate experiments and it should be fully annotated, meaning that experts determined the diseases that are or are not present as well as annotated the individual heart beats. ECG databases fulfill these requirements. One of the largest repositories of ECG data and physiological data is PhysioNet. PhysioNet was founded in 1999 by the National Institutes of Health (USA) and offers large collections of freely accessible ECG data [30]. These datasets vary in their size from around 10 recordings [41] to over 100 [42]. The QT Database [42] (available at https://physionet.org/content/qtdb/1.0.0/) has annotations for all types of ECG waves (P, QRS, T, and U; see Figure 1.1) for 105 two-lead recordings, each 15 minutes long. This database focuses on wave and feature detection as most ECG datasets only have the QRS complex annotated. The St Petersburg INCART 12-lead Arrhythmia Database (available at https://physionet.org/

content/incartdb/1.0.0/) contains 75 30-minute recordings that contain all 12 ECG leads. The significance of this database is that it contains all 12 ECG leads, while most ECG databases only contain 2 (see [29, 42])–this makes it possible to test multivariate detection methods as well as realistic circumstances, where a raw ECG would most likely contain 12 leads. The European ST-T Database [43] (available at https://physionet.org/content/edb/1.0.0/) contains 90 recordings of 79 subects, each being 2 hours long and containing two leads. This database is focused on the ST segment and the T wave (hence the name) and thus focuses on ischaemia detection. One of the most used databases in the literature is the MIT-BIH (Massachusetts Institute of Technology-Beth Israel Hospital) Arrhythmia Database (see [7, 8, 11, 24, 39, 40, 44]). This database is focused on arrhythmia detection and contains 48 two-lead recordings that are each 30 minutes long.

2 METHODS

TODO fix sectioning and verbal toc

This section details the methods used in this paper to investigate its hypothesis: does the use of the MSAX representation improve the performance of the HOT SAX anomaly detection algorithm applied to ECGs compared to the SAX representation? TODO fix this and make congruent with hypothesis. The following section will first cover the mathematical foundations of SAX, MSAX, and HOT SAX. Then, the statistical methods used to analyze the results will be presented, followed by a note on the implementation of the mathematical methods and statistical analysis, and the data used in this research. Lastly, the limitations of these methods will be discussed. TODO make sure this fits the actual structure

While MSAX and SAX both are time series representation methods, they can be applied to ECGs, as EGCs are discrete multivariate time series. Mathematically, a discrete time series is a series of T observations made at discrete points in time, with n values recorded at each moment in time. Following [2],

$$\{\mathbf{v}[t]\}_{t\in\{1,\ldots,T\}} \tag{2.1}$$

is a n-variate time series where, for each time point t,

$$\mathbf{v}[t] = (v_1[t], \ldots, v_n[t])^T \tag{2.2}$$

represents the values of the time series. If the time series has $n = 1$ values at each time point, it is called univariate, if $n > 1$, it is called multivariate. For ECGs, the discrete points in time are dictated by the sampling frequency, which is the number of observations made in one second. The number of leads in an ECG is equivalent to the variable n in (2.2). As virtually all ECGs consist of more than one lead ($n > 1$), ECGs are multivariate time series.

2.0.1 SAX

The Symbolic Aggregate Approximation, introduced in 2003 by Lin, Keogh, Lonardi, and Chiu, is a symbolic time series representation [3]. Its main features are the symbolic representation and dimension reduction of time series data, and the lower bounding of the Euclidean Distance. A lower bound (or infimum) in set theory is a value that is the largest element in a set S that is smaller than all elements in a certain subset of S. For SAX, lower bounding the Euclidean Distance can be understood as stating that the SAX distance between two SAX representations is guaranteed to be smaller than or equal to the "true" or Euclidean Distance between the original time series. Accordingly, the distance between two SAX representations is guaranteed to be representative of the Euclidean Distance between the raw time series. This feature sets SAX apart from other symbolic time series representations, and, together with its wide use in the literature [3, 14, 17, 18, 19, 21, 23, 45, 46, 47, 48, 49, 50, 51] and application to ECGs [24], makes SAX a promising methods to use. The SAX representation only works for time series $\mathbf{v}[t]$ for which $n = 1$, i.e. which are univariate. Thus (2.1) becomes $\mathbf{v}[t] = v_1[t]$. Using the SAX representation is a three-step process. Firstly, the raw time series is normalized. Secondly, the dimension of the normalized time series is reduced using PAA. Thirdly, the PAA-represented time series is discretized. Additionally, a distance measure between two SAX representations is defined.

Normalization. The normalization for the SAX representation is necessary because, to compare two time series, it is standard practice to normalize both of them because otherwise comparisons between them are not useful [3]. SAX is normalized by applying standard Z-normalization, resulting in a time series with sample mean equal to 0 and sample standard deviation equal to 1. To do this, the mean and standard deviation of the univariate time series $\mathbf{v}[t]$ needs to be calculated. The sample mean of a list of values is

$$\overline{x} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{v}[t].$$

The sample standard deviation can be found with the formula

$$s = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(\mathbf{v}[t]-\overline{x})^2}$$

(It should be noted that for applications to whole ECGs, the sample standard deviation and population standard deviation are very similar, as T is often > 100.000). Finally, the normalized time series values can be obtained by computing

$$\mathbf{v}[t] = \frac{\mathbf{v}[t]-\overline{x}}{s}, \qquad \forall t \in \{1,\dots,T\}.$$

The resulting time series will have the same shape as the raw time series, but it will have no unit and be normalized.

TODO insert figure here

08:paa-proc

Dimension reduction with PAA. The dimension reduction of the SAX representation is due to the use of PAA. The PAA method takes a univariate time series $\mathbf{v}[t]$ of length T and an integer w and segments $\mathbf{v}[t]$ into w segments, taking the average of each. Following [3], the resulting representation is denoted as $\overline{\mathbf{v}}[t]$ and now has length w. The PAA representation of $\mathbf{v}[t]$ can be calculated by using the following formula [3]

$$\overline{\mathbf{v}}[t] = \frac{w}{T}\sum_{j=\frac{n}{w}(t-1)+1}^{\frac{n}{w}t}\mathbf{v}[t], \qquad \forall t \in \{1,\dots,w\}. \tag{2.3}$$

08:eq

Now $\mathbf{v}[t]$ has been converted to the PAA representation $\overline{\mathbf{v}}[t]$. This process reduces the length of the time series from T to w, with the dimension reduction ratio depending on the choice of w.

TODO insert figure here

Table 2.1: Breakpoint values for numbers of breakpoints a from 3 to 6. The parameter a determines into how many equally-sized areas the normal curve $\mathcal{N}(0,1)$ is split. The breakpoints β_i delimit the areas. Table contents are quoted from [3].

a / β_i	3	4	5	6
β_1	-0.43	-0.67	-0.84	-0.97
β_2	0.43	0	-0.25	-0.43
β_3	—	0.67	0.25	0
β_4	—	—	0.84	0.43
β_5	—	—	—	0.97

Discretization of PAA representation. This last step in the SAX representation process involves transforming the PAA representation $\overline{\mathbf{v}}[t]$ into a sequence of equiprobable symbols. Here it is assumed that a normalized time series has a Gaussian normal distribution ($\mathcal{N}(0,1)$). The number symbols used is denoted by a–the alphabet size. To create the equiprobable symbols, Lin, Keogh, Lonardi, and Chiu [3] use so-called "breakpoints". These breakpoints are a sorted list of numbers $B = \beta_1, \ldots, \beta_{a-1}$. The area under the normal curve $\mathcal{N}(0,1)$ (i.e. the probability) between two consecutive segments β_i and $\beta_{i+1} = 1/a$. This creates a segments ($a-1$ breakpoints) of $\mathcal{N}(0,1)$ that have the same area, i.e. the same probability. The values of the breakpoints in B can be found in a Z-table. For illustration, Table 2.1 shows the breakpoint values for $a = 3$ to $a = 6$.

Once the breakpoint values have been determined, the discretization process begins. The process assigns all PAA segments whose value is below β_1 the symbol "a". The PAA segments falling in the area $\beta_1 \leq$ and $< \beta_2$ are assigned "b". This mapping process is continued, until all PAA segments are symbolized. Now we have arrived at the SAX representation. The SAX representation of $\overline{\mathbf{v}}[t]$ is denoted $\hat{\mathbf{v}}[t]$ and has the same length as $\overline{\mathbf{v}}[t]$ (w). Mathematically, the discretization process is formulated in [3] as

$$\hat{\mathbf{v}}[t] = \mathrm{alpha}_j \quad \text{if } \beta_{j-1} \leq \hat{\mathbf{v}}[t] < \beta_j, \qquad \forall t \in \{1, \ldots, w\}.$$

Here alpha_j is the jth letter of the alphabet, i.e. $\mathrm{alpha}_1 =$ "a", $\mathrm{alpha}_2 =$ "b" … The resulting time series representation has an even more reduced dimension than PAA because instead of infinitely many possible values for the real-valued PAA values, now there are only a different, equiprobable symbols. Thus, the SAX representation $\hat{\mathbf{v}}[t]$ has been obtained. TODO insert graph here

SAX distance measure. A distance measure between two SAX representations of the same length is required to be able to compare them with each other. The SAX distance function is based on the Euclidean Distance between two time series $\mathbf{v}[t]$ and $\mathbf{u}[t]$ is [3]

$$\mathrm{D}\left(\mathbf{u}[t], \mathbf{v}[t]\right) \equiv \sqrt{\sum_{t=1}^{T} \left(\mathbf{u}[t] - \mathbf{v}[t]\right)^2}.$$

Table 2.2: A table for the dist function for $a = 5$. Each cell displays the distance between the symbols denoting its row and column. The formula for the cell values is (2.5).

	a	b	c	d	e
a	-0.43	-0.67	-0.84	-0.97	
b	0.43	0	-0.25	-0.43	
c	0	0.67	0.25	0	
d	0	0	0.84	0.43	
e	0	0	0	0.97	

Through the PAA distance as an intermediate step, the authors arrive at MINDIST in (2.4), the SAX distance function that returns the minimum distance between the two original time series. It is defined as [3]

$$\text{MINDIST}\left(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t]\right) \equiv \sqrt{\frac{T}{w}}\sqrt{\sum_{t=1}^{w}\left(\text{dist}(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t])\right)^2}. \tag{2.4}$$

The function dist is based on a lookup table that contains the distances between two symbols. Table 2.2 shows the lookup table for $a = 5$. The values of each table cell are 0 for symbols letters or the absolute difference of the breakpoints otherwise. The formula

$$\text{cell}_{r,c} = \begin{cases} 0, & \text{if } |r - c| \leq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & \text{otherwise} \end{cases} \tag{2.5}$$

is used to calculate the values of each cell in Table 2.2 by r (row) and c (column) [3]. **TODO** inset example of distance between a short segment of SAX stuff, maybe two ecg segments

2.0.2 MSAX

The Multivariate Symbolic Aggregate Approximation was introduced by Anacleto, Vinga, and Carvalho in 2020. It is an extension of SAX to multivariate time series [2]. It shares the main features of SAX, but expands them to multivariate time series, such as ECGs–n can be any integer ≥ 1. A lower bound for the MSAX distance function also exists, i.e. distance between two MSAX representations is, just as in SAX, guaranteed to be representative of the Euclidean Distance between the raw time series. As MSAX builds on the legacy of SAX and purports to improve upon it, it makes a good research topic. Further, having only been introduced in 2020, MSAX is new and there is still much to be learned about it and its applications. The very similar performance the authors observed between SAX and MSAX in ECG applications motivate further research in this area as they note in their conclusion [2]. Using the MSAX representation has the same steps as SAX: normalization, PAA-based dimension reduction, and discretization. A variation of the MINDIST function exists, too.

Normalization. The rationale for normalization in the MSAX representation is twofold. Firstly, the same considerations as for SAX apply with regards to comparing two time series. Secondly, MSAX utilizes multivariate normalization to take advantage of the covariance structure of multivariate time series data. To avoid confusion with the previous section, a multivariate time series shall be denoted as $\mathbf{V}[t]$. Multivariate normalization relies on a sample mean vector containing the sample mean for each of the time series $(V_1[t], \ldots, V_n[t])^T$ in $\mathbf{V}[t]$. The sample standard deviation is replaced by a covariance matrix. The sample mean vector is equivalent to the vector of expected values $\vec{E}$, following anacleto2020 [2]:

$$E(\mathbf{V}[t]) = \vec{E} = \begin{bmatrix} \mathrm{mean}(V_1[t]) \\ \vdots \\ \mathrm{mean}(V_n[t]) \end{bmatrix} = \begin{bmatrix} \frac{1}{T}\sum_{t=1}^{T} V_1[t] \\ \vdots \\ \frac{1}{T}\sum_{t=1}^{T} V_n[t] \end{bmatrix}.$$

The covariance matrix, an $n \times n$ matrix, contains the variance of each part $(V_1[t], \ldots, V_n[t])^T$ of $\mathbf{V}[t]$ on its main diagonal, and the covariance between ith and jth parts of $\mathbf{V}[t]$ in the (i, j) position. The general form of a covariance matrix is shown in 08:eq:cov-mat (2.6) below. The covariance matrix is denoted as $\mathrm{Var}(\mathbf{V}[t])$ or $\Sigma_{n\times n}$. It is calculated as:

$$\mathrm{Var}(\mathbf{V}[t]) = \Sigma_{n\times n} = \begin{bmatrix} \mathrm{cov}(V_1, V_1) & \ldots & \mathrm{cov}(V_1, V_n) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(V_n, V_1) & \ldots & \mathrm{cov}(V_n, V_n) \end{bmatrix}. \tag{2.6}$$

08:eq

The covariance of two time series parts $V_i[t]$ and $V_j[t]$ is defined as the mean of product of the difference between the values of $V_i[t]$ and its expected value. The following equation illustrates this process:

$$\mathrm{cov}(V_i[t], V_j[t]) = E\Big(\big[V_i[t] - E(V_i[t])\big] \cdot \big[V_j[t] - E(V_j[t])\big]\Big)$$

(Note that $\mathbf{V}[t]$ can be conceptualized as a matrix, with its row representing the different sub-series and the columns representing specific values of t). Once $\vec{E}$ and $\Sigma_{n\times n}$ have been found, the time series $\mathbf{V}[t]$ can be normalized by the following formula anacleto2020 [2]:

$$\mathbf{V}[t] = (\Sigma_{n\times n})^{-1/2}\left(\mathbf{V}[t] - \vec{E}\right).$$

The result will have a mean of zero and uncorrelated variables anacleto2020 [2].

TODO insert figure here, use both leads in both graphs; may turn out a little large

Dimension reduction with PAA. Dimension reduction using PAA for MSAX is performed in exactly the same way as for SAX. The procedure outlined in the previous section is applied to each of the elements $(V_1[t], \ldots, V_n[t])^T$ of the time series $\mathbf{V}[t]$–equation 08:eq:paa (2.3) is applied to each part. This results in a PAA representation of the original time series $\overline{\mathbf{V}}[t] = \left(\overline{V_1}[t], \ldots, \overline{V_n}[t]\right)^T$. This process also reduces the length of the time series from T to w for each sub-series, with the dimension reduction ratio depending on the choice of w anacleto202 [2] TODO insert figure here

Discretization of PAA representation. The discretization of the PAA representation for MSAX also works like it does for SAX. Like in the previous paragraph, the process used in the SAX representation is applied to each of the sub-time series in $\overline{\mathbf{V}}[t]$ to obtain $\widehat{\mathbf{V}}[t]$. The alphabet size a is the same for each $V_n[t]$ and the symbols are found in the same way as in the SAX representation. The breakpoint values are calculated the same and Table 08:tab:B 2.1 is as valid for MSAX as it is for SAX. The assigning of symbols is generally performed in the same way, too. For bivariate time series ($n = 2$), $V_1[t]$ could be assigned lowercase symbols ("a", "b" …) while $V_2[t]$ could be assigned uppercase symbols ("A", "B" …). This has no impact on the method, it is simply a visual aid for the viewer to distinguish the values. The final MSAX representation $\widehat{\mathbf{V}}[t]$ will consist of one long list of symbols because for each moment t all generated symbols are combined into a list for this time that represent all sub-time series at that time. TODO add a graph and example here for illustration

MSAX distance measure. The MSAX distance measure expands MINDIST to multivariate time series. This is done by adding an additional summarization step to the MINDIST function. The MSAX distance MINDIST_MSAX operates on two MSAX representations $\widehat{\mathbf{U}}[t], \widehat{\mathbf{V}}[t]$. Both representations must have the same length w and same number n. MINDIST_MSAX sums the distances between the individual elements $U_i[t], V_i[t]$ for $i = \overline{1, \dots, n}$. The following equations expresses MINDIST_MSAX anacleto2020 [2].

$$\text{MINDIST\_MSAX}\left(\widehat{\mathbf{U}}[t], \widehat{\mathbf{V}}[t]\right) \equiv \sqrt{\frac{T}{w}} \sqrt{\sum_{t=1}^{w} \left(\sum_{i=1}^{n} \left(\text{dist}(\widehat{U}_i[t], \widehat{V}_i[t]) \right)^2 \right)}. \qquad (2.7)$$

08:eq

The function dist is the same as the SAX function, being based on equation 08:eq:dist-cell (2.5) and lookup tables like Table 08:tab:dist 2.2. Like MINDIST, MINDIST_MSAX also lower bounds the Euclidean Distance and derives all the same benefits from that. TODO inset example of distance between a short segment of SAX stuff, maybe two ecg segments

2.0.3 HOT SAX

The Heuristically Ordered Time series using Symbolic Aggregate Approximation is a discord discovery algorithm introduced by keogh2005 Keogh, Lin, and Fu in keogh2005 2005 keogh2005 [21]. Discord discovery is the process of identifying subsections of a time series that are most different to other segments of the time series, i.e. that have the largest distance to other, non-intersecting subsegments keogh2005 [21]. The standard approach to discord discovery, comparing all segments to all other segments, is too slow for application to large datasets as it has a complexity of $O(m^2)$. This means that for m subsegments, around m^2 operations need to be performed. This would be performed in two nested loops, the outer loop iterating over all subsegments. The inner loop also iterates over all subsegments; the subsegments from the outer and inner loop are compared if and only if they are not identical. An algorithm for this procedure can be found in Table 1 of keogh2005 [21]. As each of these loops would iterate over all of the m subsegments, resulting in the mentioned m^2 complexity.

HOT SAX has the goal of speeding up this process and making discord discovery viable even for long time series. The authors theorize that a "magic" heuristic would provide the time series subsegments first in order of their distance to their nearest neighbor, from largest to smallest. These would be iterated over by the outer loop. Then, the magic heuristic would provide an ordering of the subsegments by their distance to the subsegment selected in the outer loop, in ascending order. Inside the inner loop, the subsegments are them compared, given that they are not identical. The logic behind the outer and inner magic heuristics is as follows: the outer heuristic orders the time series subsegments by their distance to their neighboring segments, descendingly. This effectively produces a list of subsegments that are most different from the other segments. Combined with the assumption that time series discords are very different from the other segments, the outer heuristic effectively orders the subsegments by the likelihood that they are discords. The inner heuristic returns an ordering that produces subsegments with the smallest distances to the outer-loop subsegment, i.e. it returns the segments most similar to the outer-loop subsegment. If it is assumed that the outer-loop subsegment is likely a discord, other subsegments that are similar to it are also likely to be discords. With these two magic heuristics, discord discovery would be sped up significantly, as the discords are very likely found early on in the process and thus the process can be abandoned before exhausting all m^2 operations. Even if the magic heuristic were as bad as possible, returning orderings that slow down the process as much as possible, the brute force method mentioned in the previous paragraph would not be faster. In this case, both methods would require m^2 operations and be equal [**keogh2003**].

The magic heuristic can of course not exist, hence the name. But keogh2005 Keogh, Lin, and Fu approximate it to still achieve a significant speedup. The first step the authors take is to apply the SAX representation to the time series to reduce its complexity and dimension, while retaining an accurate representation of the data lin2003 [3]. To compare two subsegments, the SAX distance function MINDIST is used. Then, a certain window size is chosen that represents the number of SAX segments that will make up one of the aforementioned time series subsegments. Now the magic heuristic can be approximated. The outer heuristic, which returns the subsegments of the time series in descending order by their distance to their neighbors. The authors approximate this by taking each SAX subsegment and insert them into an array, counting how many times each unique time series occurs. By sorting this array by the occurrence counts, the outer heuristic can be approximated. At the same time as the outer heuristic, the inner heuristic can be approximated. For its approximation, a digital tree (also known as trie or prefix tree) is used. In this tree, the SAX representation is used as an index to locate a leaf node. This node contains the locations in the time series where the particular subsequence occurs. Effectively, this prefix tree can be used to locate all SAX subsequences that are identical to a given one. So, if one as the subsegment "abc", it is possible to find all other locations of the "abc" subsegment in the time series using this tree. **TODO** use Figure 4 from [keogh2005] here By default, HOT SAX only returns the most discordant time series subsegment. Because more than one time series discord can be present in a time series (as in an ECG), it is useful to extract more than one. The number of discords to be extracted is called k. If $k > 1$, each newly found discord is compared to the other already found discords, and only the k discords with the largest distance values are saved. In this way, it is possible to extract an exact number of discords.

Another modification is to save all discords that the method detects [21].

The authors find that HOT SAX can effectively detect time series discords and speed up the process when compared to the brute force method. A strength of the method is, in their opinion, the fact that it only requires the user to determine a single parameter, the length of the subsequence (the parameter k has no influence on the method itself, it simply determines how many of the results should be used) [21]. The authors apply HOT SAX to ECG time series and, in anecdotal tests, find success in determining discords in ECGs. They further suggest the application of HOT SAX to multivariate time series [21]. For these reasons, HOT SAX was chosen as this research's discord discovery method. While HOT SAX was designed to work with SAX, all it requires is a time series representation and a lower-bounding distance measure defined on it. MSAX exhibits both of those traits and can thus be used with HOT SAX. Doing this expands HOT SAX to multivariate time series, as MSAX can represent those. Using HOT SAX with the MSAX representation is novel and a contribution of this research.

3 RESULTS

TODO fix up this section, set proper headers, etc TODO add in some information on how much data, which parameter combinations, etc were used

3.1 Implementation

This subsection is concerned with the implementation of the methods discussed in the previous section. TODO spell out what will be covered in which order

All code used in the implementation of these methods is available upon request via email at

`konarski_m@auca.kg`

.

How I implemented the above stuff. Languages, approaches, hurdles, all the details needed to reproduce this research. Also mention the simplifications I chose to make and why: no sliding window, only even divisors, only divisors within sampling frequency and cutting ECG to even multiple of sampling frequency.

3.1.1 The ECG Data

The ECG data used in this research is the MIT-BIH Arrhythmia Database moody2001 goldberger2000 [29, 30]. This database contains 48 ECG recordings that are each 30 minutes long. For each of the ECGs, this database contains two ECG leads and is thus multivariate data. The leads chosen are not the same in each ECG, they were chosen based on which of the 12 originally recorded ones best represent the condition of the ECG. A team of cardiologists annotated each heartbeat in each ECG and determined if it is a normal heartbeat or not. For example, a beat with the annotation "N" is a normal beat, while "A" denotes an atrial premature beat. The annotations also include non-beat features such as a change in signal quality, denoted by "~", or a rhythm change, which is denoted as "+". A full list of annotations and their meaning is available at https://archive.physionet.org/physiobank/annotations.shtml. These annotations make it possible to judge the performance of a discord detection algorithm, as each detected discord can be checked for correctness using the provided annotations. While the MIT-BIH database is not the only database that possesses such annotations, it is one of the most commonly used ones in the literature (see [illegible] [7, 8, 11, 24, 39, 40, 44]) and it represents a middle ground in a couple important respects. The databases 48 ECGs are a manageable number, falling in between the extremes of around 10 and over 100 ECGs. Furthermore, the 30 minutes length represent real-world ECGs better than 10 second excerpts, but are not as long and analysis-intensive as 24 hour recordings. Lastly, the MIT-BIH database has a sampling frequency of 360 samples per second, which is an adequate value kligfield2007 [38]. The ECG data in this database is unfiltered.

The ECG data can be downloaded using the PhysioNet website at the url https://www.physionet.org/content/mitdb/1.0.0/, or, alternatively, using the PhysioNet-developed WFDB applications package. This package provides command line applications to work with PhysioNet data. For each of the individually numbered ECG records, 4 files exist. The

`.hea`

files contain metadata on the ECG record, including anonymized patient information and the lead names. The

`.dat`

files contain the actual ECG recording and the other two files contain additional information, including the annotations. Once the ECG recording has been downloaded, the

`rdsamp`

command is used to convert the binary ECG recording files into a more user-friendly comma separated value (CSV) file. The

`rdann`

command is then used to create a CSV file containing the annotations for each of the ECG records. Finally, the ECG recording data and the annotations can be merged into a single file by using the time stamps contained in both files. This yields full ECG recordings with added beat annotations in one file. These files are the basis of all further methods and analysis performed in this research. The author created a script in the Julia programming language that performs this process.

3.1.2 SAX, MSAX, HOT SAX Implementation

The main program for this research was developed using the Julia programming language. Julia is a scientific programming language that has similarities to R, MATLAB, and Python. Julia possesses a rich ecosystem of libraries for visualization, computation, and data manipulation. For more information, visit the Julia website at https://julialang.org/. The following subsection will detail the steps comprising the discord discovery program.

The first step is the selection of the important parameters for the methods. The user defined parameters are:

- the sampling frequency of the ECG data to be analyzed;
- the number of PAA segments w used for SAX and MSAX;
- the alphabet size a used for SAX and MSAX;
- the subsequence length that determines HOT SAX;
- the variable k indicating how many discords should be found.

These parameters determine all actions the program performs afterward. The second step is to load a CSV file containing the ECG data and annotations into the program. Once the ECG file is loaded, it is transformed into a data frame. A data frame is a type of data structure that can hold heterogeneous data types, e.g. text and numbers. This step adds important information to the ECG data. The ECG data frame contains the parameters itemized above to enable reproduction and analysis of the results, an index range for each PAA segment so it can be located in the raw ECG,

Table 3.1: Contingency table showing the relationship between detected discords and actual annotated values.

08:ta

Assigned Actual	Discord Detected	Non-Discord Detected
Is Discord	True Positive	False Negative
Is Non-Discord	False Positive	True Negative

the beat annotations for each PAA segment, and empty data fields for the results of the analysis with HOT SAX. The next step is the application of the SAX and MSAX representations. The transformation of the raw time series data to the symbolic representations is performed in the same order as discussed earlier in this section, and thanks to the Julia programming language's ecosystem of libraries, can be easily translated into code. SAX is applied to each of the ECG leads individually, while MSAX is applied as designed to both at once. HOT SAX comprises the next step. For MSAX, the HOT SAX process is performed using the MSAX representation and distance measure. The method returns a list of distances as well as a list of indices that indicate which PAA segment has which distance. Depending on the parameter k, only the top k of these discords are returned. These results are then added to the respective PAA segments in the ECG data frame, adding both the MSAX distance of the segment as well as a binary indicator of whether or not the segment was detected as a discord. For SAX the process slightly different. Because SAX is a univariate representation, it cannot be directly applied to a bivariate ECG. Thus, SAX is applied to each lead of the ECG separately and HOT SAX is performed for each representation of each lead. Each set of results is, like MSAX, a list of indices of PAA segments and a list of their distances. Each sets of results is also added to the ECG data frame. This time the detection indicator is quaternary, it represents no detection, detection on the first lead, detection on the second lead, or detection on both leads. After both of these processes are completed, the ECG data frame is written to a CSV file for further analysis. This process can be repeated thousands of times to create data of different values for the parameters to determine optimal values and their influence.

3.1.3 Statistical Analysis of Results

After completing the computations for different sets of parameters, the results need to be analyzed. While HOT SAX is not a classifier in the sense of classifying heartbeats by medical standards, it does classify them into discords and non-discords. Thus, it is a binary classifier. Binary classifiers can be evaluated using the well-known True Positive, True Negative, False Negative, and False Positive values. Table 08:tab:bin-class 3.1 shows their relationship. The values in Table 08:tab:bin-class 3.1 can be used to calculate many useful ratios that assist the evaluation of the HOT SAX algorithm. This research uses the recall value (also known as sensitivity), the accuracy, and the precision. These ratios are calculated as follows: recall value is defined as

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},$$

the accuracy as

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}},$$

and the precision as

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

Recall can be understood as a measure of how many of the actual discords were correctly assigned the label discord. This is the most important measure for the analysis of HOT SAX applied to ECGs because in a medical scenario, identifying as many possibly relevant sections of the ECG is more important than being 100% accurate in their identification. The second most important value is precision, which can be understood as a measure of how many of the detected discords are actually discords. While it is more important to identify as many discords as possible, a 100% recall rate could be achieved simply by assigning the label of discord to every element in the time series. Furthermore, detecting too many non-discords as discords makes it harder to analyze the actual discords that were highlighted. This of course is not useful, and thus the precision of HOT SAX needs to be incorporated into the analysis. Lastly, accuracy is not a very good measure for this particular application, as the majority of the segments in an ECG are non-discords and HOT SAX only detects a minority of the segments in an ECG. This leads to a high True Negative rate and thus a relatively high accuracy, even if HOT SAX did not actually detect any actual discords. Nonetheless, accuracy is a very common indicator of classifier performance and will thus be considered.

The analysis of the methods was performed using the data whose generation was discussed above. The analysis was performed using the R programming language. R is an established statistical and mathematical programming language with great support for statistical methods and tests. The first step in the analysis was the processing of the data generated using the Julia program. This consisted of calculating the True Positive, True Negative, False Negative, and False Positive values for each parameter combination and each method. A segment was considered a "non-discord" if its annotation consisted of an "N" or nothing "". The former is obvious; the decision to consider no annotation ("") a non-discord was made because for certain segments of the ECGs, no annotations were available. This can happen if, for example, the subsequence length for HOT SAX is much smaller than one heartbeat. In that situation, one heartbeat might be represented by 5 or more subsegments. The heartbeat annotation, given for a specific point in time, will only fall into one of the 5 segments and can thus only be counted for that one segment. The same is true for an annotation showing a discord. This method of analysis puts HOT SAX at a disadvantage because a discord located in one subsegment might influence its neighboring segments and thus lead to their detection. This detection might be an actual discord being detected, but counting it as one would incorrectly inflate the True Positive rate by assuming something about the data that it itself does not support without some inference. Thus the decision was made to accept lower True Positive values than may be accurate. TODO explain why? or do so later in limitations section. Any annotation that was not empty or "N" was considered a discord. This includes the medical annotation for arrhythmia but also the annotations for changes in signal quality or noise. This done because HOT SAX is not meant to classify heartbeats by medical significance, but by how different they are from other heartbeats. A very noisy normal heartbeat will be detected the same as a arrhythmic beat. The classification of the detected discords into medically normal and abnormal heartbeats is left to more sophisticated analysis methods or human experts. The purpose of the HOT SAX methods is merely to reduce the number of ECG segments that need to be analyzed by pre-selecting the beats likely to contain useful information. After calculating the True Positive, True Negative, False Negative, and False Positive value for each parameter combination, they were collected in a data frame also containing information on the parameters that lead to them. These data frames are then saved as CSV files for further analysis. The contingency values were analyzed for the HOT SAX with MSAX method, for HOT SAX with individual SAX (only considering a single ECG lead), and for a HOT SAX with combined SAX method where the detected discords of the individual HOT SAX with SAX computations were combined. The very last step in the analysis was the calculation of the average recall, precision, and accuracy across all 48 ECGs for SAX and MSAX. This allows for a simpler comparison of the results for different parameters and enables pruning of certain parameter combinations before more sophisticated analysis begins.

REFERENCES

rockwell2016 [1] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, en, ser. Springer Texts in Statistics. Cham: Springer International Publishing, 2016, ISBN: 978-3-319-29852-8 978-3-319-29854-2. DOI: 10.1007/978-3-319-29854-2.

anacleto2020 [2] M. Anacleto, S. Vinga, and A. M. Carvalho, "MSAX: Multivariate Symbolic Aggregate Approximation for Time Series Classification," en, in *Computational Intelligence Methods for Bioinformatics and Biostatistics*, P. Cazzaniga, D. Besozzi, I. Merelli, and L. Manzoni, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 90–97, ISBN: 978-3-030-63061-4. DOI: 10.1007/978-3-030-63061-4_9.

lin2003 [3] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," en, in *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery - DMKD '03*, San Diego, California: ACM Press, 2003, pp. 2–11. DOI: 10.1145/882082.882086.

abozorgi2015 [4] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – A decade review," en, *Information Systems*, vol. 53, pp. 16–38, Oct. 2015, ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007.

shieh2008 [5] J. Shieh and E. Keogh, "*I* SAX: Indexing and mining terabyte sized time series," en, in *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08*, Las Vegas, Nevada, USA: ACM Press, 2008, p. 623, ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401966.

amahatana2005 [6] C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A Novel Bit Level Time Series Representation with Implication of Similarity Search and Clustering," en, in *Advances in Knowledge Discovery and Data Mining*, T. B. Ho, D. Cheung, and H. Liu, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2005, pp. 771–777, ISBN: 978-3-540-31935-1. DOI: 10.1007/11430919_90.

kaur2016 [7] I. Kaur, R. Rajni, and A. Marwaha, "ECG Signal Analysis and Arrhythmia Detection using Wavelet Transform," en, *Journal of The Institution of Engineers (India): Series B*, vol. 97, no. 4, pp. 499–507, Dec. 2016, ISSN: 2250-2106, 2250-2114. DOI: 10.1007/s40031-016-0247-3.

prasad2018 [8] B. V. P. Prasad and V. Parthasarathy, "Detection and classification of cardiovascular abnormalities using FFT based multi-objective genetic algorithm," en, *Biotechnology & Biotechnological Equipment*, vol. 32, no. 1, pp. 183–193, Jan. 2018, ISSN: 1310-2818, 1314-3530. DOI: 10.1080/13102818.2017.1389303.

panuccio2002 [9] A. Panuccio, M. Bicego, and V. Murino, "A Hidden Markov Model-Based Approach to Sequential Data Clustering," en, in *Structural, Syntactic, and Statistical Pattern Recognition*, G. Goos *et al.*, Eds., vol. 2396, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 734–743, ISBN: 978-3-540-44011-6 978-3-540-70659-5. DOI: 10.1007/3-540-70659-3_77.

corduas2008 [10] M. Corduas and D. Piccolo, "Time series clustering and classification by the autoregressive metric," en, *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 1860–1872, Jan. 2008, ISSN: 01679473. DOI: 10.1016/j.csda.2007.06.001.

nygaard1998 [11] R. Nygaard and D. Haugland, "Compressing ECG signals by piecewise polynomial approximation," en, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 3, Seattle, WA, USA: IEEE, 1998, pp. 1809–1812, ISBN: 978-0-7803-4428-0. DOI: 10.1109/ICASSP.1998.681812.

zhu2018 [12] H. Zhu, Y. Pan, K.-T. Cheng, and R. Huan, "A lightweight piecewise linear synthesis method for standard 12-lead ECG signals based on adaptive region segmentation," en, *PLOS ONE*, vol. 13, no. 10, J. Zhao, Ed., e0206170, Oct. 2018, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0206170.

zifan2006 [13] A. Zifan, M. H. Moradi, S. Saberi, and F. Towhidkhah, "Automated Segmentation of ECG Signals using Piecewise Derivative Dynamic Time Warping," en, p. 5, 2006.

zan2016 [14] C. T. Zan and H. Yamana, "An improved symbolic aggregate approximation distance measure based on its statistical features," in *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*, ser. iiWAS '16, New York, NY, USA: Association for Computing Machinery, Nov. 2016, pp. 72–80, ISBN: 978-1-4503-4807-2. DOI: 10.1145/3011141.3011146.

sun2014 [15] Y. Sun *et al.*, "An improvement of symbolic aggregate approximation distance measure for time series," en, *Neurocomputing*, vol. 138, pp. 189–198, Aug. 2014, ISSN: 09252312. DOI: 10.1016/j.neucom.2014.01.045.

yu2019 [16] Y. Yu *et al.*, "A Novel Trend Symbolic Aggregate Approximation for Time Series," en, vol. abs/1905.00421, p. 9, 2019. [Online]. Available: http://arxiv.org/abs/1905.00421.

lkhagva2006 [17] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: Extension of Symbolic Aggregate Approximation for Financial Time Series Data Representation," en, in *Proceeding of IEICE the 17th Data Engineering Workshop*, Ginowan, Japan, 2006, p. 7. [Online]. Available: https://www.researchgate.net/publication/229046404_Extended_SAX_extension_of_symbolic_aggregate_approximation_for_financial_time_series_data_representation (visited on 02/27/2021).

alinowski2013 [18] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, "1d-SAX: A Novel Symbolic Representation for Time Series," en, in *Advances in Intelligent Data Analysis XII*, A. Tucker, F. Höppner, A. Siebes, and S. Swift, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 273–284, ISBN: 978-3-642-41398-8. DOI: 10.1007/978-3-642-41398-8_24.

fuad2010 [19] M. M. M. Fuad and P.-F. Marteau, "TOWARDS A FASTER SYMBOLIC AGGREGATE APPROXIMATION METHOD:" en, in *Proceedings of the 5th International Conference on Software and Data Technologies*, University of Piraeus, Greece: SciTePress - Science and and Technology Publications, 2010, pp. 305–310, ISBN: 978-989-8425-22-5 978-989-8425-23-2. DOI: 10.5220/0003006703050310.

camerra2010 [20] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh, "iSAX 2.0: Indexing and Mining One Billion Time Series," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, Dec. 2010, pp. 58–67. DOI: 10.1109/ICDM.2010.124.

keogh2005 [21] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence," en, in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, TX, USA: IEEE, 2005, pp. 226–233, ISBN: 978-0-7695-2278-4. DOI: 10.1109/ICDM.2005.79.

park2020 [22] H. Park and J.-Y. Jung, "SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining," en, *Expert Systems with Applications*, vol. 141, p. 112 950, Mar. 2020, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2019.112950.

ordonez2008 [23] P. Ordóñez *et al.*, "Visualizing Multivariate Time Series Data to Detect Specific Medical Conditions," *AMIA Annual Symposium Proceedings*, vol. 2008, pp. 530–534, 2008, ISSN: 1942-597X. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2656052/ (visited on 03/30/2021).

zhang2019 [24] C. Zhang *et al.*, "Anomaly detection in ECG based on trend symbolic aggregate approximation," en, *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 2154–2167, 2019, ISSN: 1547-1063. DOI: 10.3934/mbe.2019105.

alghatrif2012 [25] M. AlGhatrif and J. Lindsay, "A brief review: History to understand fundamentals of electrocardiography," en, *Journal of Community Hospital Internal Medicine Perspectives*, vol. 2, no. 1, p. 14 383, Jan. 2012, ISSN: 2000-9666. DOI: 10.3402/jchimp.v2i1.14383.

fye1994 [26] W. Fye, "A History of the origin, evolution, and impact of electrocardiography," en, *The American Journal of Cardiology*, vol. 73, no. 13, pp. 937–949, May 1994, ISSN: 00029149. DOI: 10.1016/0002-9149(94)90135-X.

meek2002 [27] S. Meek and F. Morris, "Introduction. I—Leads, rate, rhythm, and cardiac axis," *BMJ : British Medical Journal*, vol. 324, no. 7334, pp. 415–418, Feb. 2002, ISSN: 0959-8138. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1122339/ (visited on 05/21/2021).

becker2006 [28] D. E. Becker, "Fundamentals of Electrocardiography Interpretation," *Anesthesia Progress*, vol. 53, no. 2, pp. 53–64, 2006, ISSN: 0003-3006. DOI: 10.2344/0003-3006(2006)53[53:FOEI]2.0.CO;2.

moody2001 [29] G. Moody and R. Mark, “The impact of the MIT-BIH Arrhythmia Database,” en, *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May-June/2001, ISSN: 07395175. DOI: 10.1109/51.932724.

ldberger2000 [30] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” en, *Circulation*, vol. 101, no. 23, Jun. 2000, ISSN: 0009-7322, 1524-4539. DOI: 10.1161/01.CIR.101.23.e215.

asilewski2012 [31] J. Wasilewski and L. Poloński, “An Introduction to ECG Interpretation,” en, in *ECG Signal Processing, Classification and Interpretation*, A. Gacek and W. Pedrycz, Eds., London: Springer London, 2012, pp. 1–20, ISBN: 978-0-85729-867-6 978-0-85729-868-3. DOI: 10.1007/978-0-85729-868-3_1.

xie2020 [32] L. Xie *et al.*, “Computational Diagnostic Techniques for Electrocardiogram Signal Analysis,” en, *Sensors*, vol. 20, no. 21, p. 6318, Nov. 2020, ISSN: 1424-8220. DOI: 10.3390/s20216318.

kadish2001 [33] A. H. Kadish *et al.*, “ACC/AHA Clinical Competence Statement on Electrocardiography and Ambulatory Electrocardiography,” *Circulation*, vol. 104, no. 25, pp. 3169–3178, Dec. 2001. DOI: 10.1161/circ.104.25.3169.

zelevitch2011 [34] C. Antzelevitch and A. Burashnikov, “Overview of Basic Mechanisms of Cardiac Arrhythmia,” *Cardiac electrophysiology clinics*, vol. 3, no. 1, pp. 23–45, Mar. 2011, ISSN: 1877-9182. DOI: 10.1016/j.ccep.2010.10.012.

nowbar2019 [35] A. N. Nowbar *et al.*, “Mortality From Ischemic Heart Disease: Analysis of Data From the World Health Organization and Coronary Artery Disease Risk Factors From NCD Risk Factor Collaboration,” en, *Circulation: Cardiovascular Quality and Outcomes*, vol. 12, no. 6, Jun. 2019, ISSN: 1941-7713, 1941-7705. DOI: 10.1161/CIRCOUTCOMES.118.005375.

ihd2010 [36] Institute of Medicine (US) Committee on Social Security Cardiovascular Disability Criteria, “Ischemic Heart Disease,” en, in *Cardiovascular Disability: Updating the Social Security Listings*, Washington, DC: National Academies Press (US), 2010. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK209964/ (visited on 05/21/2021).

herring2006 [37] N. Herring, “ECG diagnosis of acute ischaemia and infarction: Past, present and future,” en, *QJM*, vol. 99, no. 4, pp. 219–230, Feb. 2006, ISSN: 1460-2725, 1460-2393. DOI: 10.1093/qjmed/hcl025.

kligfield2007 [38] P. Kligfield *et al.*, “Recommendations for the Standardization and Interpretation of the Electrocardiogram: Part I: The Electrocardiogram and Its Technology: A Scientific Statement From the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society *Endorsed by the International Society for Computerized Electrocardiology*,” en, *Circulation*, vol. 115, no. 10, pp. 1306–1324, Mar. 2007, ISSN: 0009-7322, 1524-4539. DOI: 10.1161/CIRCULATIONAHA.106.180200.

sivaraks2015 [39] H. Sivaraks and C. A. Ratanamahatana, “Robust and Accurate Anomaly Detection in ECG Artifacts Using Time Series Motif Discovery,” en, *Computational and Mathematical Methods in Medicine*, vol. 2015, pp. 1–20, 2015, ISSN: 1748-670X, 1748-6718. DOI: 10.1155/2015/453214.

alupadasu2012 [40] R. Valupadasu and B. R. R. Chunduri, “Identification of Cardiac Ischemia Using Spectral Domain Analysis of Electrocardiogram,” en, in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom: IEEE, Mar. 2012, pp. 92–96, ISBN: 978-1-4673-1366-7 978-0-7695-4682-7. DOI: 10.1109/UKSim.2012.22.

baim1986 [41] D. S. Baim *et al.*, “Survival of patients with severe congestive heart failure treated with oral milrinone,” en, *Journal of the American College of Cardiology*, vol. 7, no. 3, pp. 661–670, Mar. 1986, ISSN: 07351097. DOI: 10.1016/S0735-1097(86)80478-8.

laguna1997 [42] P. Laguna, R. Mark, A. Goldberg, and G. Moody, “A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG,” en, in *Computers in Cardiology 1997*, Lund, Sweden: IEEE, 1997, pp. 673–676, ISBN: 978-0-7803-4445-7. DOI: 10.1109/CIC.1997.648140.

taddei1992 [43] A. Taddei *et al.*, “The European ST-T database: Standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography,” en, *European Heart Journal*, vol. 13, no. 9, pp. 1164–1172, Sep. 1992, ISSN: 1522-9645, 0195-668X. DOI: 10.1093/oxfordjournals.eurheartj.a060332.

kanani2020 [44] P. Kanani and M. Padole, “ECG Heartbeat Arrhythmia Classification Using Time-Series Augmented Signals and Deep Learning Approach,” en, *Procedia Computer Science*, Third International Conference on Computing and Network Communications (CoCoNet’19), vol. 171, pp. 524–531, Jan. 2020, ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.04.056.

aremu2019 [45] O. O. Aremu, D. Hyland-Wood, and P. R. McAree, “A Relative Entropy Weibull-SAX framework for health indices construction and health stage division in degradation modeling of multivariate time series asset data,” en, *Advanced Engineering Informatics*, vol. 40, pp. 121–134, Apr. 2019, ISSN: 1474-0346. DOI: 10.1016/j.aei.2019.03.003.

guigou2017 [46] F. Guigou, P. Collet, and P. Parrend, *Anomaly Detection and Motif Discovery in Symbolic Representations of Time Series*. Apr. 2017. DOI: 10.13140/RG.2.2.20158.69447.

he2020 [47] Z. He, S. Long, X. Ma, and H. Zhao, “A Boundary Distance-Based Symbolic Aggregate Approximation Method for Time Series Data,” en, *Algorithms*, vol. 13, no. 11, p. 284, Nov. 2020. DOI: 10.3390/a13110284.

lahcioglu2021 [48] B. Kulahcioglu, S. Ozdemir, and B. Kumova, “Application of Symbolic Piecewise Aggregate Approximation (PAA) Analysis to ECG Signals,” Mar. 2021.

liu2018 [49] M. Liu and Y. Kim, “Classification of Heart Diseases Based On ECG Signals Using Long Short-Term Memory,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2018, pp. 2707–2710. DOI: 10.1109/EMBC.2018.8512761.

pham2010 [50] N. D. Pham, Q. L. Le, and T. K. Dang, “HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery,” en, in *Intelligent Information and Database Systems*, N. T. Nguyen, M. T. Le, and J. Świątek, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2010, pp. 113–121, ISBN: 978-3-642-12145-6. DOI: 10.1007/978-3-642-12145-6_12.

tayebi2011 [51] H. Tayebi *et al.*, “RA-SAX: Resource-Aware Symbolic Aggregate Approximation for Mobile ECG Analysis,” in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, Jun. 2011, pp. 289–290. DOI: 10.1109/MDM.2011.67.