American University of Central Asia

Applied Mathematics and Informatics Program

# Multivariate Symbolic Aggregate Approximation for ECG Analysis

Moritz M. Konarski

A Thesis Submitted to the Applied Mathematics and Informatics Program of American University of Central Asia in Partial Fulfillment of the Requirements for the Degree of **Bachelor of Arts**

Author
**Moritz M. Konarski**

Certified by Thesis Supervisor
**Professor Taalaibek M. Imanaliev**

Accepted by
**Sergey N. Sklyar**
Head of Applied Mathematics and Informatics Program, AUCA

May 25, 2021
Bishkek, Kyrgyz Republic

# ABSTRACT

The electrocardiogram (ECG) is the most common diagnostic tool used for heart diseases. Because heart diseases claim more lives each year than any other disease, proper analysis of ECGs is very important. For humans, learning how to read an ECG is very difficult and even with a lot of training, mistakes are common. One proposed solution to this problem is computerized ECG analysis which aims to use the power of computers to improve and speed up the analysis of ECGs and to save lives. But even computers have their limits, and working with large ECG recordings can take a prohibitively long time. One approach to speeding up the analysis process is to transform the ECG into a more compact form–a representation. These representations can be used to more efficiently store and compare two ECGs. The Symbolic Aggregate Approximation (SAX) is a prominent representation method that has been modified to properly represent time series that record more than one value for each moment in time, e.g. ECGs. This modification is known as Multivariate SAX (MSAX). Besides representing ECGs, SAX can also be used as part of an algorithm called HOT SAX, which can find discords in ECGs. ECG discords are segments of the ECG that are very different from the other segments. Because they are so different, they are likely to be indicators of disease and thus their discovery can help a cardiologist save lives.

This research compares the use of MSAX with the HOT SAX algorithm (HOT MSAX) to the use of standard HOT SAX. The novel contributions of this research are to use MSAX as the representation for HOT SAX, and the application of MSAX to ECG discord discovery. Because MSAX is meant to represent time series like ECGs and SAX is not, the recall value for HOT MSAX-based ECG discord discovery should be greater than the recall value for the HOT SAX-based method. Through experimental testing using the MIT-BIH Arrhythmia Database it is shown that, when each method is used with its optimal parameters, there is no significant difference between HOT SAX and HOT MSAX. Nonetheless, HOT MSAX performs as well as HOT SAX and could thus find applications in ECG discord discovery by pre-selecting potentially relevant ECG segments for the cardiologist to analyze and thus speeding up and simplifying the diagnosis process.

**Keywords**: Symbolic Aggregate Approximation, ECG Discord Discovery, HOT SAX, MSAX, Time Series Analysis

# ACKNOWLEDGMENTS

This independent research is partly supported by the Faculty Research Grant “Mathematical Model in Acute Cardiac Ischemia Evaluation” of the American University of Central Asia.

I want to thank my supervisor, Professor Taalaibek M. Imanaliev, for his invaluable help and feedback. I would also like to thank Bektur Daniyarov for his insight into the underlying theory of ECGs and their analysis.

# TABLE OF CONTENTS

**Introduction** . . . . . **5**
**1 Background and Related Work** . . . . . **7**
1.1 Time Series and Time Series Analysis . . . . . 7
1.1.1 Time Series Representation Methods . . . . . 8
1.1.2 SAX Representation Method . . . . . 9
1.2 ECGs and ECG Analysis . . . . . 10
1.2.1 What is an ECG? . . . . . 10
1.2.2 Computerized ECG analysis . . . . . 12
1.2.3 ECG databases . . . . . 14
**2 Methods** . . . . . **15**
2.1 SAX . . . . . 15
2.2 MSAX . . . . . 20
2.3 HOT SAX . . . . . 23
2.4 Data . . . . . 25
2.5 Implementation . . . . . 26
2.6 Statistical Analysis Methods . . . . . 27
**3 Results** . . . . . **29**
3.1 ECG Analysis Program . . . . . 29
3.1.1 ECG Preprocessing . . . . . 29
3.1.2 SAX, MSAX, HOT SAX, and HOT MSAX . . . . . 29
3.1.3 Computerized Statistical Analysis . . . . . 30
3.1.4 Limitations of the ECG Analysis Program . . . . . 31
3.2 Data Analysis . . . . . 32
3.2.1 Dataset 1 . . . . . 32
3.2.2 Dataset 2 . . . . . 33
3.2.3 Comparison of Optimal Parameters . . . . . 39
**4 Discussion** . . . . . **41**
**5 Conclusion** . . . . . **43**
**References** . . . . . **44**

# INTRODUCTION

Heart diseases are the most deadly diseases on the planet, with 16% of all annual global deaths being caused by ischemic heart disease [1]. Ischemic heart disease is a condition caused by restricted blood flow to an area of the heart which causes the heart muscle to not receive enough blood. A restriction can be caused by a blood clot or by plaque buildup and if the flow of blood is restricted too much, it can cause myocardial ischemia. Myocardial ischemia, commonly known as a heart attack, is the result of oxygen-deprived heart tissue dying [2]. Fortunately, ischemic heart disease can be diagnosed before it causes a heart attack. One method of diagnosis involves a stress test during which the heart's activity is recorded using an electrocardiograph. The resulting recording reflecting the heart's activity is called an electrocardiogram (ECG or EKG) [2].

The ECG is a diagnostic tool used to evaluate patients with (suspected) heart problems. It is a non-invasive, real-time, and cost-effective method that may be used to diagnose IHD and other heart diseases like arrhythmia (the presence of irregular heartbeat rhythms). The ECG is the most common tool used for cardiac analysis and diagnosis [3, 4, 5]. The electric measurements which an ECG records are taken in millivolts, which represent the electrical activity of the heart with each heartbeat. Figure 1 shows the electrical activity an ECG records for a single heartbeat. This is also called the ECG wave [4, 5]. When a disease affects the way the heart beats, an ECG can record those changes and they can then be used to diagnose the disease.

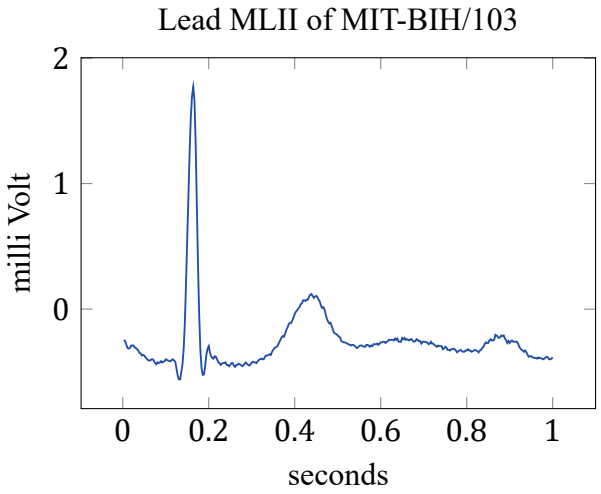


Figure 1: An ECG wave.

Unfortunately, using an ECG to diagnose a cardiac condition is difficult. The changes caused by diseases are often small and can be easily missed. Furthermore, ECGs can be longer than 24 hours, and analyzing such a large amount of data is very time-intensive [3, 5]. To mitigate this issue, computers have been used to perform ECG analysis. Computers can be used to perform some or all of the steps involved in ECG analysis, which are listed below [4]:

1. signal acquisition and filtering,
2. data transformation or preparation for processing,
3. waveform recognition,
4. feature extraction, and
5. classification or diagnosis.

Of the above steps, steps 2 and 5 are particularly interesting. Step 2 is important because, while computers can analyze data faster than humans, they, too, have trouble analyzing large amounts of data [6]. As part of the data transformation step, the complexity and size of the ECG data can

be reduced. The methods used for this step are called time series representation methods [7]. An established and widely used time series representation is the Symbolic Aggregate Approximation (SAX). It is possible to apply time series representation methods to ECGs because ECGs are time series. A time series is a set of values recorded at specific times [8] and ECGs are a special type of time series–a multivariate time series. Multivariate time series record multiple values for each point in time [9], for ECGs these values are the different leads. There are also time series representation methods for multivariate time series. Multivariate SAX is an extension of SAX to multivariate time series that incorporates the relationships between the different elements, e.g. the ECG leads, and should make MSAX a better representation than SAX for multivariate data [9].

ECG analysis step 5 traditionally involves a cardiologist manually looking at the heartbeats of an ECG to determine if they are normal or if the heartbeat represents an anomaly [10]. A type of time series anomaly are discords. Discords are segments of a time series that are very different from the other segments of the time series. These discords can represent important features of time series, e.g. possibly abnormal heartbeats in an ECG. Accordingly, it is desirable to quickly find the discords in a time series. This is called classification, each segment of the ECG is either classified as a discord or as a non-discord. While this can be done by comparing all segments of a time series to all other segments, HOT SAX presents a more efficient solution. HOT SAX is a discord discovery algorithm that uses the SAX representation and intelligent heuristics to speed up the discord discovery process [11]. While HOT SAX uses the SAX representation, the only requirements of the algorithm are that two representations can be compared to each other. As MSAX is such a representation [9], HOT SAX could be used with MSAX.

This work aims to investigate how the use of the MSAX representation combined with the HOT SAX algorithm, called HOT MSAX hereinafter, will influence ECG discord discovery performance compared to the standard HOT SAX algorithm and which parameters achieve the best performance. The HOT SAX and HOT MSAX algorithms will be compared experimentally using a well-known ECG database called the MIT-BIH Arrhythmia Database [12, 13]. The performance of the algorithms will be primarily measured by their recall (also known as sensitivity). Recall is relevant here because when it comes to discovering potentially medically relevant discords, it is better to be a bit too diligent and mark too many segments, lowering the accuracy, if it means that more discords can be discovered. A robust and fast ECG discord discovery system could significantly speed up and simplify the work of a cardiologist by automatically classifying segments of an ECG into discords and non-discords. The discords could then be analyzed first and, if the method is sufficiently accurate, present the cardiologist with the medically relevant segments very quickly. Classifying ECG anomalies with the HOT MSAX algorithm should achieve a greater recall value than with HOT SAX because, unlike SAX, MSAX is designed to work with multivariate data like ECGs.

The following paper first contains a section reviewing the background and work related to time series, ECGs, and their analysis. Secondly, this work's methods, SAX, MSAX, and HOT SAX, are covered in detail. The third section presents the results of this research. Fourth is a discussion of the results, followed by a conclusion.

# 1 BACKGROUND AND RELATED WORK

This section provides background information on time series and ECGs, as well as methods to analyze them. First, time series analysis will be covered, focusing on different time series representation methods and the SAX representation. Then, ECGs and their analysis will be discussed, focusing on what an ECG is, computerized ECG analysis, and ECG databases.

## 1.1 Time Series and Time Series Analysis

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 [8]. 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 [9]. 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 [6, 7]:

- 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 [6]. 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 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 [7, 14], there are four types of dimension reduction methods:

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

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 [7].

### 1.1.1 Time Series Representation Methods

**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 [7]. This representation simply transforms the raw time series into a sequence of 1s and 0s. A data point is assigned a 1 if its value 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 [15].

**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) [7]. 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 [16], 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 its input signal into many sinus waves of different frequencies. In [17] 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 section 2.1.

**Model-based representation** Model-based methods use stochastic methods such as Hidden Markov Models (HMM) and the Auto-Regressive Moving Average (ARMA) [7]. An HMM was used in [18] to cluster electroencephalograph recordings (measuring the brain's electrical activity). It was found that their method 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 [19].

**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), and SAX [7]. PPA can be used to compress an ECG by approximating it using polynomials. With second-order polynomials, ECGs can be compressed with a minimal level of distortion [20]. The authors of [21] use a modified PLA representation with adaptive ECG segmentation to successfully reconstruct the 12 standard leads of an ECG from only 3 leads. The SAX representation will be covered in detail in section 2.1. The following subsection 1.1.2 will provide background on the method and its variations.

### 1.1.2 SAX Representation Method

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 [22] and a measure of the trend of each averaged segment [23, 24]. Extended SAX modifies SAX to include the minimum and maximum values of each segment for improved representation of the raw data [25] while 1d-SAX incorporates a linear regression over each segment into SAX [26]. A combination of SAX and a polynomial approximation was used to speed up the SAX method [27]. To improve the indexing performance of SAX, iSAX introduced convertible alphabet sizes, allowing SAX representations with different alphabet sizes to be compared with each other and indexed into a tree structure [14]. 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 [28]. 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 [11]. For an in-depth description of this method, please refer to section 2.3.

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 [29]. 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 the 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 accurate, and in the remain-

ing 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 [9]. MSAX will be thoroughly presented in section 2.2. Another application of SAX to multivariate data used it to visualize multivariate medical test results and enable their analysis [30]. 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 [31], 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 in 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 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 2). 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 [3, 32]. 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 [33].

### 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 heartbeat. 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 [10]. Figure 2 shows an ECG wave of a single heartbeat from record 103 of the MIT-BIH database [12, 13] (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 [10, 34]. The last P wave is part of the next heartbeat, which is not shown in Figure 2.

The waves and complexes shown in Figure 2 are the object of ECG analysis. Changes

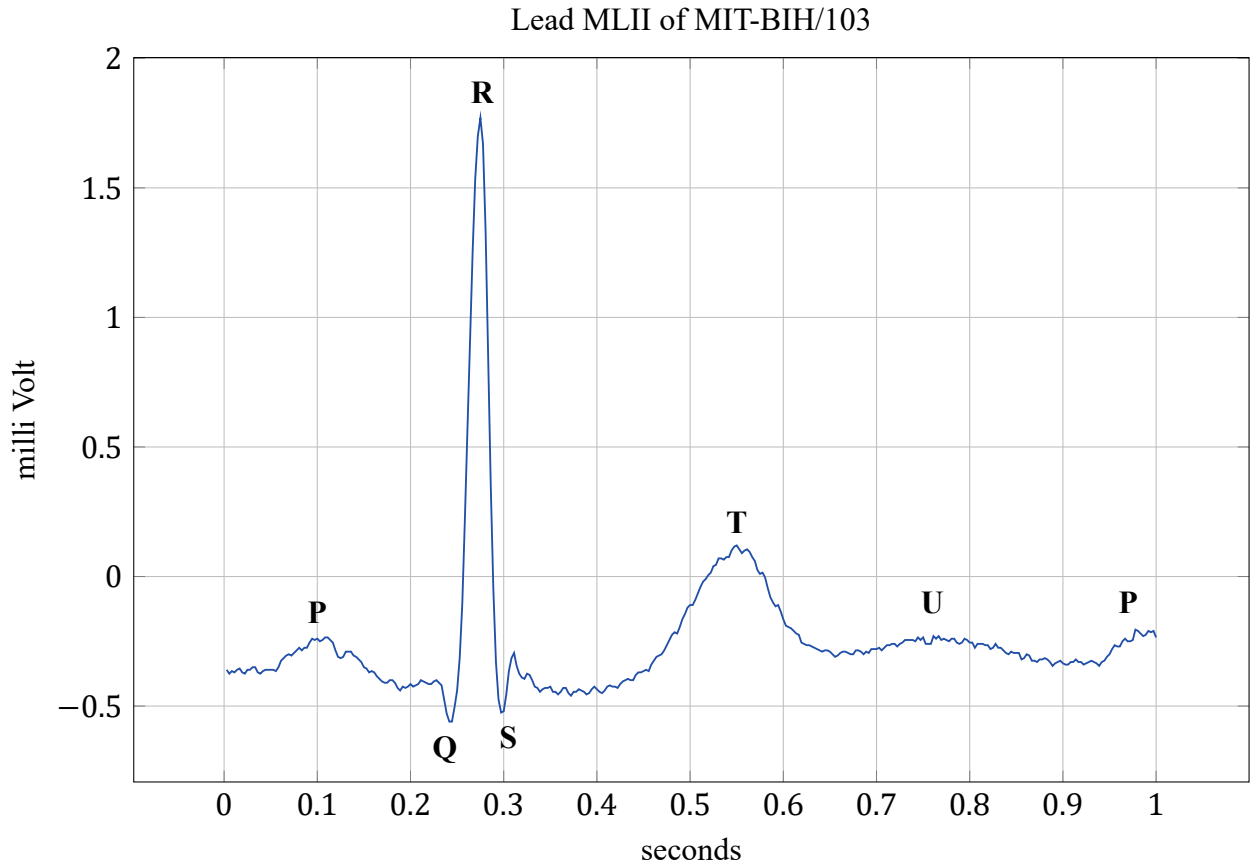


Figure 2: Annotated ECG of one heartbeat. This graph is based on lead II, data points 2031–2390 of recording 103 of the MIT-BIH database [12, 13].

in their shape, duration, or height can indicate heart conditions. Becker list some of the features relevant for ECG analysis [10]:

- 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 [3, 5]. 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 diagnoses 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 [35].

Two types of heart conditions that an ECG can detect are cardiac arrhythmias and ischemic 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 [36]. In an ECG, arrhythmia is most apparent in changes in the interval between the QRS complexes. Ischemic heart disease is the main cause of death worldwide [37]. Ischemic heart disease is characterized by restricted blood flow to an area of the heart, causing it to not receive enough blood and oxygen, as mentioned in the introduction. In an ECG, ischemic heart disease can be diagnosed based on changes in the ST segment and the T wave. The diagnosis of ischemic 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 [38].

### 1.2.2 Computerized ECG analysis

The widespread use of ECGs and the time-sensitive nature of their application as diagnostic tools make 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 [4]. Time series analysis methods can also be applied to ECGs because ECGs simply represent discrete multivariate time series. As discussed in section 1.1, 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 [4], are:

1. signal acquisition and filtering,
2. data transformation,
3. waveform recognition,
4. feature extraction, and
5. classification or diagnosis.

Step 1 comprises the digital recording of ECG signals or the digitizing of paper-based ECG records. For either process, 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 [4]. 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 afterward. Choosing appropriate wavelet parameters can be challenging, but methods

to optimize this process have been proposed [5]. Step 2 uses the same types of methods for dimension reduction that were discussed in the above sections for time series and shall not be repeated here. This includes SAX, which has been successfully applied to ECG analysis [31]. 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 2. The Fast Fourier Transform provides a way of analysing the frequency domain of the ECG signal, enabling the detection of the QRS complex [17] 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 [16]. 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 [5].

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 [16]. Other algorithms focus on separating classes of data, pattern recognition, or retaining the structure of the original data [5]. Here, again, the time series representation methods discussed in sections above can be applied.

Finally, the extracted features can be classified; this is stage 5. In this stage, judgments 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 but which is very expensive to calculate for high-dimensional data. Support vector machines are used for pattern recognition and can work with small samples. Artificial neural networks are robust and can work with complex problems, they are generally more accurate than support vector machines [17]. 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 and perform the analysis steps internally, as a single model [5]. This approach is relatively new and still actively researched. The previous approach, too, is enjoying active research attention.

### 1.2.3 ECG databases

A very important element of computerized 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 heartbeats. 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 [13]. 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 2) 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 [12, 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 subjects, 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 ischemia detection. One of the most used databases in the literature is the MIT-BIH (Massachusetts Institute of Technology-Beth Israel Hospital) Arrhythmia Database (see [16, 17, 20, 31, 39, 40, 44]). This database is focused on arrhythmia detection and contains 48 two-lead recordings that are each 30 minutes long.

# 2 METHODS

This section details the methods used in this work to investigate its research question and test its hypothesis. Below, a short introduction to time series will be given. Following that, the SAX representation will be discussed, followed by the MSAX representation, and then the HOT SAX algorithm. The ECG data used to experimentally test this work's hypothesis is discussed, followed by a note on the implementation of the methods, and finally the statistical methods used for ECG analysis.

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 [9],

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

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

$$\mathbf{v}[t] = (v_1[t], \dots, v_n[t])^T \tag{2}$$

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). As virtually all ECGs consist of more than one lead ($n > 1$), ECGs are multivariate time series.

## 2.1 SAX

The Symbolic Aggregate Approximation, introduced in 2003 by Lin, Keogh, Lonardi, and Chiu, is a symbolic time series representation [6]. 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 (see [6, 11, 22, 25, 26, 27, 30, 45, 46, 47, 48, 49, 50, 51]) and application to ECGs (see [31]), makes SAX a promising method to use. The SAX representation only works for time series $\mathbf{v}[t]$ for which $n = 1$, i.e. which are univariate. Thus (1) becomes $\mathbf{v}[t] = v_1[t]$. Applying 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 the Piecewise Aggregate Approximation (PAA). Thirdly, the PAA-represented time series is discretized. Additionally, a distance measure between two equal-length 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 [6]. 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

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

The sample standard deviation can be found with the formula

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

(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

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

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

**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 [6], the resulting representation is denoted as $\bar{\mathbf{v}}[t]$ and now has length $w$. The PAA representation of $\mathbf{v}[t]$ can be calculated by using the following formula [6]

$$\bar{\mathbf{v}}[t] = \frac{w}{T} \sum_{j=\frac{n}{w}(t-1)+1}^{\frac{n}{w}t} \mathbf{v}[t], \qquad \forall t \in \{1, \dots, w\}. \tag{3}$$

Now $\mathbf{v}[t]$ has been converted to the PAA representation $\bar{\mathbf{v}}[t]$. This process reduces the length of the time series from $T$ to $w$, with the dimension reduction ratio depending on the choice of $w$. In Figure 3, the PAA representation is shown overlaid onto ECG 103 of the MIT-BIH database. It is apparent that the dimension of the ECG has been reduced. The original ECG section of one second contains 360 data points, while its PAA representation only contains 18 values–a dimension reduction of 20. The parameter $w$ is one of only two parameters that determine the SAX representation. A low number of parameters can be a great advantage in time series analysis because it is simpler

to optimize a few, understandable parameters compared to many unintuitive parameters.

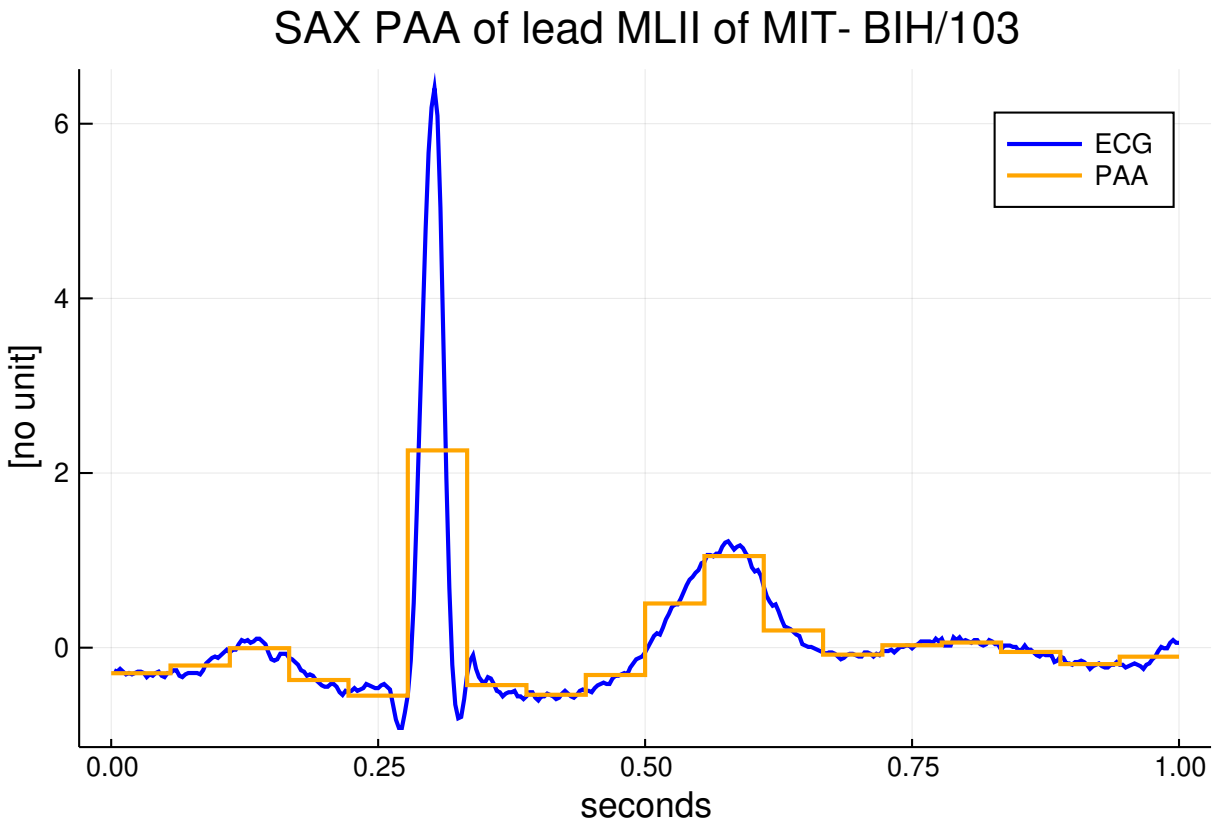


Figure 3: Graph of ECG 103 of the MIT-BIH database overlaid with its PAA representation. Here $w = 18$, meaning there are 18 PAA segments. The dimension reduction through PAA is 20.

**Discretization of PAA representation.** This last step in the SAX representation process involves transforming the PAA representation $\bar{\mathbf{v}}[t]$ 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 [6] 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 $\beta_i$ 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 1 shows the breakpoint values for $a = 3$ to $a = 6$. This parameter is the second and last parameter that influences the SAX representation.

Table 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 $\beta_i$ delimit the areas. Table contents are quoted from [6].

| $\beta_i$ \ $a$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| $\beta_1$ | -0.43 | -0.67 | -0.84 | -0.97 |
| $\beta_2$ | 0.43 | 0 | -0.25 | -0.43 |
| $\beta_3$ | — | 0.67 | 0.25 | 0 |
| $\beta_4$ | — | — | 0.84 | 0.43 |
| $\beta_5$ | — | — | — | 0.97 |

Once the breakpoint values have been determined, the discretization process begins. The process assigns all PAA segments whose value is below $\beta_1$ 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 $\bar{\mathbf{v}}[t]$ is denoted $\hat{\mathbf{v}}[t]$ and has the same length as $\bar{\mathbf{v}}[t]$ $(w)$. Mathematically, the discretization process is formulated in [6] as

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

Here $\text{alpha}_j$ is the $j$th letter of the alphabet, i.e. $\text{alpha}_1$ = "a", $\text{alpha}_2$ = "b" … The resulting time series representation has even lower complexity 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. In Figure 4 ECG 103 of the MIT-BIH database is shown with $w = 18$ and $a = 4$. The three breakpoints are indicated by the dashed horizontal lines. Figure 4 illustrates how an ECG can be reduced from 360 real-valued points to 18 symbols. The SAX representation of this ECG section is "bbbbb**d**bbbc**d**cbddbbb". Both the QRS complex (the first "d") and the T wave (second "d") can be seen in the representation.

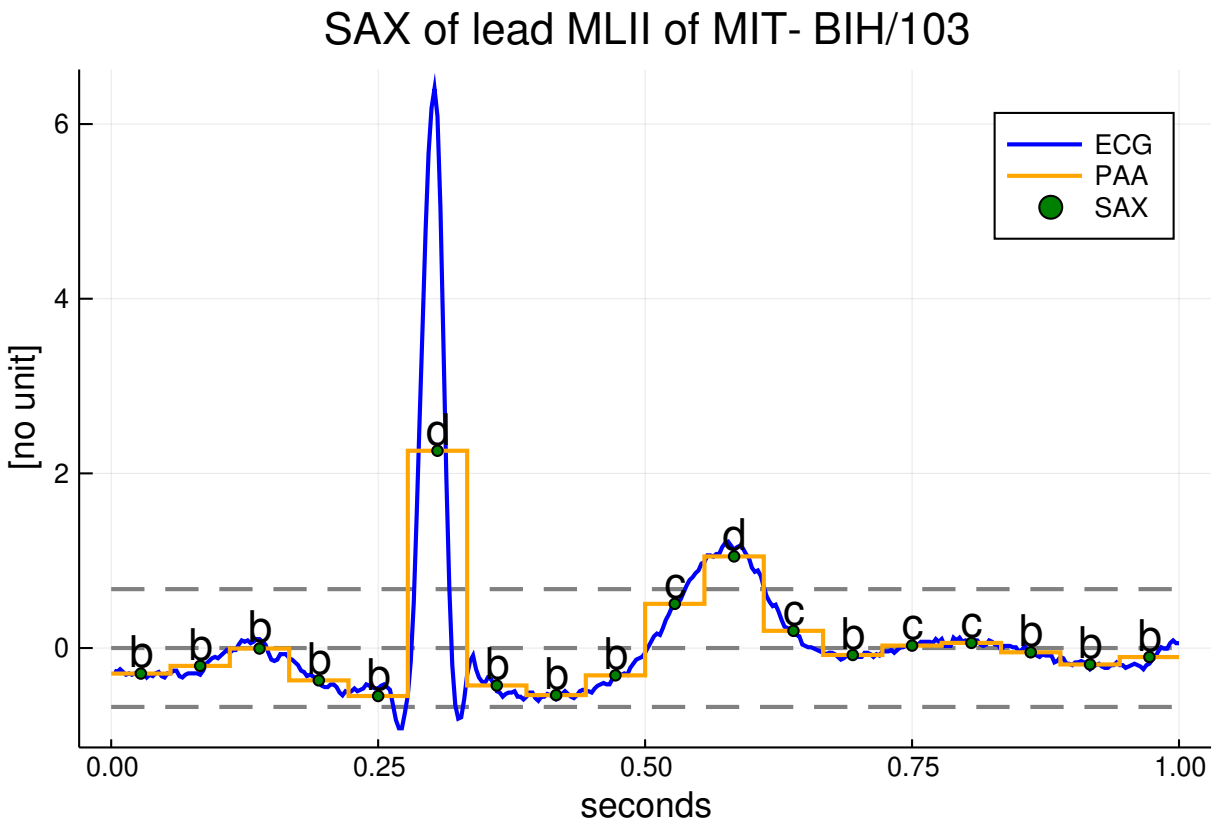


Figure 4: Graph of ECG 103 of the MIT-BIH database overlaid with its PAA representation and the SAX discretization. Here $w = 18$, a dimension reduction of 20, and $a = 4$, meaning there are 3 breakpoints (indicated by the dashed lines) and 4 symbols.

**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 [6]

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

Through the PAA distance as an intermediate step, the authors arrive at $MINDIST$ in (4), the SAX distance function that returns the minimum distance between the two original time series. It is defined as [6]

$$MINDIST\left(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t]\right) \equiv \sqrt{\frac{T}{w}} \sqrt{\sum_{t=1}^{w} \left(dist(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t])\right)^2}. \quad (4)$$

The function $dist$ is based on a lookup table that contains the distances between two symbols. Table 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

$$\text{cell}_{r,c} = \begin{cases} 0, & \text{if } |r-c| \leq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & \text{otherwise} \end{cases} \tag{5}$$

is used to calculate the values of each cell in Table 2 by $r$ (row) and $c$ (column) [6].

Table 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 (5).

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 0 | 0.59 | 1.09 | 1.68 |
| b | 0 | 0 | 0 | 0.51 | 1.09 |
| c | 0.59 | 0 | 0 | 0 | 0.59 |
| d | 1.09 | 0.51 | 0 | 0 | 0 |
| e | 1.68 | 1.09 | 0.59 | 0 | 0 |

For example, if $a = 5$, the $dist$ between "a" and "a" is 0, just like the distance between "b" and "a". The distance between "d" and "a" is 1.09.

## 2.2 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 [9]. It shares the main features of SAX but expands them to multivariate time series, such as ECGs–$n$ can be any integer $\geq 1$. 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 [9]. 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. Like SAX, MSAX only depends on two parameters, the PAA segment count $w$ and the alphabet size $a$. This makes it simple to optimize MSAX as the parameters are few and intuitive.

**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 $(V_1[t], \dots, V_n[t])^T$ 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 [9]:

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

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 $i$th and $j$th parts of $\mathbf{V}[t]$ in the $(i, j)$ position. The general form of a covariance matrix is shown in (6) below. The covariance matrix is denoted as $\text{Var}(\mathbf{V}[t])$ or $\Sigma_{n\times n}$. It is calculated as:

$$\text{Var}(\mathbf{V}[t]) = \Sigma_{n\times n} = \begin{bmatrix} \text{cov}(V_1, V_1) & \ldots & \text{cov}(V_1, V_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(V_n, V_1) & \ldots & \text{cov}(V_n, V_n) \end{bmatrix}. \quad (6)$$

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:

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

(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 [9]:

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

The result will have a mean of zero and uncorrelated variables [9].

**Dimension reduction with PAA.** Dimension reduction using PAA for MSAX is performed in 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 (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 $w$ [9]. The Figure 3, while showing the PAA representation for SAX, is also applicable here, as the process is identical for SAX and MSAX.

**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 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. Figure 5 shows the MSAX representation of one second of ECG 100 from the MIT-BIH database. As MSAX is a multivariate representation, both leads of the ECG are shown. The symbols of the second lead have been capitalized to distinguish them from the symbols of the first lead. The parameters for the graph are $w = 18$ and $a = 4$, leading to a dimension reduction of 20 and the use of 3 breakpoints (indicated by the dashed lines). Using the MSAX representation, the two ECG leads, which have 360 data points each in raw form, can be represented by 36 symbols. The ECG section displayed below can be expressed as the series of symbols "cCcCbC**dD**bCbCbCbBbAbBcCcC cBcBcBcBcBbB". In the MSAX representation, the features of the raw data are still present, which can be seen in the visible QRS complex indicated by "dD".

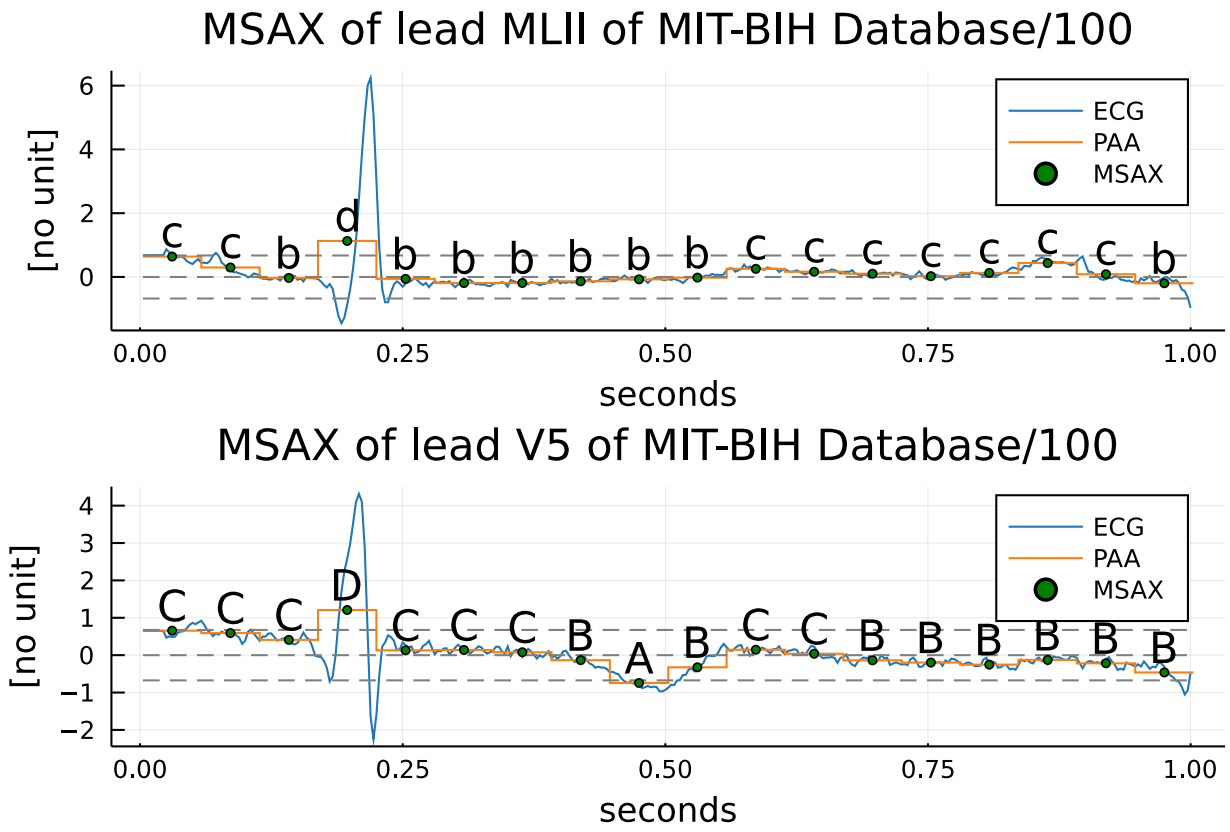


Figure 5: Graph of ECG 100 of the MIT-BIH database overlaid with its PAA representation and the MSAX discretization, for both leads. Here $w = 18$, a dimension reduction of 20, and $a = 4$, meaning there are 3 breakpoints (indicated by the dashed lines) and 4 symbols.

**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 $MINDIST\_MSAX$ [9].

$$MINDIST\_MSAX\left(\widehat{\mathbf{U}}[t], \widehat{\mathbf{V}}[t]\right) \equiv \sqrt{\frac{T}{w}} \sqrt{\sum_{t=1}^{w} \left( \sum_{i=1}^{n} \left( dist(\widehat{U}_i[t], \widehat{V}_i[t]) \right)^2 \right)}. \tag{7}$$

The function $dist$ is the same as the SAX function, being based on equation (5) and lookup tables like Table 2. Like $MINDIST$, $MINDIST\_MSAX$ also lower bounds the Euclidean Distance and derives all the same benefits from that. The distances between individual MSAX symbols the same as between SAX symbols (see Table 2). Distances between groups of more than 1 symbol can be calculated using (7).

## 2.3 HOT SAX

The Heuristically Ordered Time series using Symbolic Aggregate Approximation is a discord discovery algorithm introduced by Keogh, Lin, and Fu in 2005 [11]. 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 [11]. 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 [11]. 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 [11].

The magic heuristic can of course not exist, hence the name. But 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 [6]. 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 can be approximated by taking each SAX subsegment and inserting them into an array, counting how many times each unique time series occurs. This is shown on the left side of Figure 6. 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 has the subsegment “abc”, it is possible to find all other locations of the “abc” subsegment in the time series using this tree. A sketch of such a tree is shown on the left side of Figure 6.

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 [11].

Keogh, Lin, and Fu 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) [11]. 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 [11]. 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. The resulting algorithm is called HOT MSAX. This algorithm

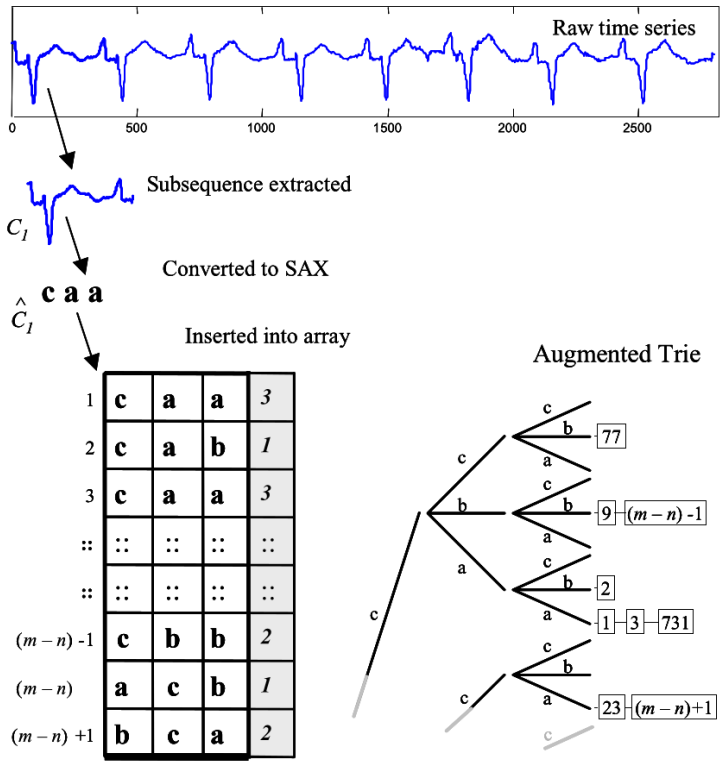


Figure 6: Illustration of the HOT SAX heuristic creation process. This figure illustrates how each unique time series subsegment is recorded in an array together with the count of its occurrence. The augmented trie stores the locations of the segments in the time series. The indices can be retrived by indexing the trie using a SAX subsequence. This figure is quoted from Figure 4 on p. 5 of [11].

expands HOT SAX to multivariate time series, as MSAX can represent those. Using HOT MSAX is novel and a contribution of this research.

**Limitations of the Methods** The limitations of the methods used in this research are the following: HOT SAX and HOT MSAX are not classifiers based on medically relevant information, they classify ECG segments into discords non-discords. The methods to not work with beat types. This means that their applications to diagnosing heart conditions are limited. HOT SAX and HOT MSAX can be used to pre-select specific ECG segments to look analyze because they exhibit discords, but it cannot, by itself, perform any type of diagnosis.

## 2.4 Data

The ECG data used in this research to evaluate the HOT SAX and HOT MSAX algorithms is the MIT-BIH Arrhythmia Database [12, 13]. This database contains 48 ECG recordings that are each 30 minutes long. For each of the ECGs, the 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 [16, 17, 20, 31, 39, 40, 44]) and it represents a middle ground in a couple of 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 [4]. The ECG data in this database is unfiltered, which adds to the real-world nature of it, as every recorded ECG contains noise of varying levels.

## 2.5 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 Julia program performs the SAX and MSAX transformation as well as the HOT SAX and HOT MSAX discord discovery and saves the result to a file for further processing. During the SAX and MSAX transformations, the parameters $w$ and $a$ define the behavior of the representations. When performing HOT SAX and HOT MSAX discord detection, the parameter $m$, the subsequence length, is considered. The parameter $k$ does not influence the actual behavior of HOT SAX or HOT MSAX, but it influences the results by controlling the number of detected discords that are returned by the methods. For each analyzed parameter set, the results and parameter values will be saved for further analysis.

The statistical analysis was performed using the R programming language, a well-known statistical programming language.

All the code used in this work is available upon request (please contact `konarski_m@auca.kg`).

**Limitations of the Implementation** The implementation of the SAX and MSAX representations only allows the use of PAA segment counts $w$ that are factors of the sampling frequency of the ECG database. This was done so that the whole ECG, being an even multiple of the sampling frequency itself, can be evenly divided into PAA segments. This decision prohibits certain numbers of PAA segments as there may numbers that are not factors of the sampling frequency but that do evenly divide the number of raw data points in the ECG. A further simplification step in the same vein is the restriction of subsequence values $m$ to numbers that evenly divide the number of PAA segments $w$. This was also done to simplify the process and to guarantee that the whole ECG would be evenly divisible into subsequences.

Table 3: Contingency table showing the relationship between detected discords and actual annotated values.

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

## 2.6 Statistical Analysis Methods

After completing the computations of HOT SAX and HOT MSAX for different sets of parameters, the results need to be analyzed. Neither HOT SAX nor HOT MSAX are classifiers in the sense of classifying heartbeats by medical standards, they do classify them into discords and non-discords. Thus, they represent binary classifiers. Binary classifiers can be evaluated using the well-known True Positive, True Negative, False Negative, and False Positive values. Table 3 shows their relationship. The values in Table 3 can be used to calculate many useful ratios that assist the evaluation of the HOT SAX and HOT MSAX algorithms. 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 and HOT MSAX 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/MSAX 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/MSAX did not detect any actual discords. Nonetheless, accuracy is a very common indicator of classifier performance and

is calculated here.

Once these statistical measures have been calculated, the parameter optimization can begin. During this process, the large set of parameters will be reduced until an optimal set of parameters $k$, $w$, $m$, and $a$ is found. The first step in this process is the introduction of a threshold. As stated above, recall is the most important statistical factor in this work. Accordingly, only parameter sets that achieve an average recall value of 95% or above for the MIT-BIH database. Only the subset of the data fulfilling this requirement will be considered. As the second most important parameter, precision is considered next. The subset of the data equal to or greater than the threshold will be sorted by average precision in descending order, singling out the parameter sets that have a recall value $\geq$ 95% and, in this subset, the highest precision. Next, the first 10 parameter sets will be chosen from this subset. Those represent a kind of "top 10" of the parameters: they fulfill the threshold and have the highest precision values. Out of these top 10 values, one optimal set of parameters should be chosen. This should be possible by creating a box plot of the 10 ten parameter sets vs their recall value. If the visual analysis does not yield concrete results, the method with the lowest spread (standard deviation or interquartile range) should be considered the optimal method. If this should prove inconclusive as well, an analysis of possible outliers, the distribution of precision values, or the distribution of the accuracy could be performed.

# 3 RESULTS

This section provides all the results obtained using the methods detailed in the section above. Firstly, the ECG analysis program will be discussed. Then, data analysis will be performed.

## 3.1 ECG Analysis Program

This subsection is concerned with the implementation of the methods discussed in the previous section. First, the ECG data and its preparation will be discussed, followed by notes on the implementation of the SAX, MSAX, and HOT SAX methods. Lastly, the process used to analyze the results is discussed. All code used in the implementation of these methods is available upon request via email at `konarski_m@auca.kg`.

### 3.1.1 ECG Preprocessing

The ECG data used in this project 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 timestamps 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 analyses performed in this research. The author created a script in the Julia programming language that performs this process. Filtering of the ECG data is not performed. The rationale behind this is twofold. Firstly, the combination of PAA and discretization in SAX and MSAX has a smoothing effect that exhibits some of the same properties as filtering. Additionally, filtering of ECGs adds many more parameters that can be modified to improve the performance of the methods, which is not desirable for this research as the methods should depend on the least possible number of parameters for simplicity. As additional support for this approach, [31] can be considered, which successfully uses SAX in their ECG analysis without mentioning any filtering performed on the ECG data.

### 3.1.2 SAX, MSAX, HOT SAX, and HOT MSAX

For the analysis to start, a set of parameters $k$, $w$, $m$, and $a$ needs to be set. Then, the next step is to load a CSV file containing the ECG data and annotations into the program. These files were generated in the preprocessing step. 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, the beat annotations for each PAA segment, and empty data fields for the results of the analysis with HOT SAX and HOT MSAX. 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 in the methods section. Thanks to the Julia programming language's ecosystem of libraries, the formulas and functions 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 and HOT MSAX are the next step. For HOT MSAX, the 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, HOT SAX is applied to each lead of the ECG separately. Each set of results is, like for HOT MSAX, a list of indices of PAA segments and a list of their distances. Each set 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 Computerized Statistical Analysis

The analysis of the methods is performed using the data whose generation was discussed above. The first step in the analysis is 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. Here, the HOT SAX and HOT MSAX methods can be segmented into three datasets. The dataset generated using HOT MSAX stays the same is from now on referred to simply as MSAX. The data generated with HOT SAX can be divided into two distinct data sets. The first one treats SAX like a univariate representation and is thus called Single SAX (S-SAX). For S-SAX, the discords which were discovered by HOT SAX on one lead are treated separately from the discords discovered on the second lead. This means that a segment is counted as a discord if it was discovered by one method or by both methods. Effectively this generates two sets of results that can be concatenated. S-SAX represents the performance of applying SAX to just one lead of the ECG. The second option is a middle ground between S-SAX and MSAX. For this dataset, a discord was counted if it was discovered by either the first or the second lead or both. As a result, SAX is, in a way, applied to a multivariate time series. Because

SAX is applied to both ECG leads, this dataset is called Dual SAX (D-SAX).

To calculate the values shown in Table 3, the meaning of a true discord and a true non-discord in the ECG needs to be defined. A segment is 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 and HOT MSAX 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 does not support without some inference. Thus the decision was made to accept lower True Positive values than may be accurate.

Any annotation that was not normal was just considered a discord. This includes the medical annotations for arrhythmia but also the annotations for changes in signal quality or noise. This is done because, as mentioned, HOT SAX and HOT MSAX are 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 an 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 and HOT MSAX 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, the values 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 three datasets mentioned above, S-SAX, D-SAX, and MSAX. The last step in the analysis was the calculation of the average recall, precision, and accuracy across all 48 ECGs for each parameter set.

### 3.1.4 Limitations of the ECG Analysis Program

The discussion of the definition of normal and abnormal heartbeats above explains why empty annotations have to be counted as normal beats and while the author believes that this is necessary, it does negatively influence the results. Another limitation is that the implementation of the SAX and MSAX representations only allows the use of PAA segment numbers that evenly divide the sampling frequency of the ECG database. This was done so that the whole ECG, being an even multiple of the sampling frequency itself, can be evenly divided into PAA segments. This decision prohibits certain numbers of PAA segments as there may be numbers that do not evenly divide into the sampling frequency but that do evenly divide the number of raw data points in the ECG. A

further simplification step in the same vein is the restriction of subsequence values to numbers that evenly divide the number of PAA segments $w$. This was also done to simplify the process and to guarantee that the whole ECG would be evenly divisible into subsequences.

## 3.2 Data Analysis

In this section, the results of the data analysis will be presented. For both the first and second datasets used in the research, the parameter selection and a summary of the results will be presented. For dataset 2, the analysis will be completed in accordance with the methods laid out in section 2.6.

### 3.2.1 Dataset 1

The first dataset is based on parameters that were arbitrarily chosen. This was done because the behavior of the methods was not yet known. While the choice was arbitrary, it was attempted to spread the values out in a reasonable way. For each parameter, Table 4 shows the method that it influences, the values that were chosen for it, and the rationale behind choosing those values. For SAX and MSAX, $w$ the number of PAA segments in one second is equivalent to dividing 360 data points by $w$. Thus, $w$ has to be a factor of 360. These factors were chosen arbitrarily and kept small because it was not known how time-intensive the computations would be. The second parameter is $a$, the alphabet size which determines the number of different symbols used in the SAX and MSAX discretization processes. Because the alphabet size is constrained by the size of the English alphabet, numbers were chosen arbitrarily in that range. For HOT SAX and HOT MSAX, the parameter $k$ is number of discords they return. Giving it the value -1 indicates that all available discords should be returned, regardless of how many there are. The values for $k$ were chosen arbitrarily. This parameter does not affect the performance of the method, it just determines how many of the results are considered. Parameter $m$ is chosen after parameter $w$ for SAX and MSAX and represents the number of PAA segments that are grouped to form a HOT SAX or HOT MSAX subsequence. This parameter must evenly divide $w$.

Table 4: Table of the methods used for dataset 1. the parameters of each method, the rationale behind the parameter choice, and the values the parameter takes are shown.

| Method | Parameter | Rationale | Values |
|---|---|---|---|
| SAX/MSAX | $w$ | arbitrary factors of 360 | 2,3,4,5,12,20,30,40,60 |
| | $a$ | arbitrary, $2 \leq a \leq 25$ | 4,5,6,7,8,9,10,12,14,17,20 |
| HOT SAX/MSAX | $k$ | arbitrary | -1,25,50,100, 150,200,300,500 |
| | $m$ | arbitrary factors of 360 and of $w$ | 2,3,4,5,12,20,30,40,60 |

Using these parameters and the programs created as part of this research, the 48 ECGs of the MIT-BIH database were analyzed using HOT SAX and HOT MSAX. The 2,640 unique sets of parameters resulting from Table 4 applied to 48 ECGs creates a dataset with 126,720 files. These files were analyzed as stated in section 2.6. The mean values of the statistical measures for each

set of unique parameters were calculated. Then, the threshold of recall ≥95% was applied to the summarized data to select the parameter combinations that yield acceptable results. Upon further analysis, it was noted that most of the parameter combinations that achieved recall ≥95% had $m = w$. To investigate this, the parameter combinations with $m \neq w$ and recall ≥95% were extracted. Table 5 shows the results of this analysis.

Table 5: Results of the analysis of dataset 1. The total number of parameter sets and the number and proportion of parameter sets in dataset 1 that fulfill the analysis conditions are presented for each method.

| Method | Total Sets | Sets Satisfying Analysis Conditions | |
|---|---|---|---|
| | | recall ≥95% | recall ≥95% and $m \neq w$ |
| S-SAX | 2,640 | 3 (0.1%) | 0 (0%) |
| D-SAX | | 13 (0.5%) | 0 (0%) |
| MSAX | | 23 (0.9%) | 3 (0.1%) |

As Table 5 shows, less than 1% of the parameter sets have a recall of ≥95%, regardless of the method used. Additionally, only 3 (0.1%) parameter sets for MSAX have the desired recall and use a value for $m$ that is different from $w$. These sets of results are too small to continue this analysis, as the goal is to analyze a "top 10" selection of values. Thus, a second dataset needs to be computed. The data presented in Table 5 does show that using subsequence lengths $m$ for the HOT SAX and HOT MSAX methods that are not equal to the PAA segment count $w$ is not effective. According to these findings, $m$ will be set equal to $w$ for the computation of dataset 2.

### 3.2.2 Dataset 2

Based on the analysis of dataset 1, the parameter selection for dataset 2 is optimized. As the value of $m$ is set equal to the value of $w$, the complexity of the computation is reduced dramatically. This is caused by two things. Firstly, the total number of parameter sets is decreased when factors $m$ of $w$ are not considered–the total number of parameter sets that have to be used is decreased. Secondly, dividing the number of PAA segments in 1 second into multiple subsegments increases the number of subsequences that HOT SAX and HOT MSAX need to work with and thus the complexity of the computation. By not doing that, the increase in complexity is avoided. As a result of this reduction in complexity, a larger set of the other parameter was considered. The parameters chosen for dataset 2 are shown in Table 6. For the SAX and MSAX parameters, all possible values were considered. Parameter $w$ can be all factors of 360. For the alphabet size $a$, all possible values were considered. The HOT SAX and HOT MSAX parameters were chosen as follows. Parameter $k$ was again assigned arbitrary values that provide a decent coverage for reasonable values. The value of -1 is again included to signify the use of all available discords. Parameter $m$ does not need to be chosen for this dataset, as it is always set to the value of $w$. All parameters used in dataset 2 have the same meaning as in dataset 1, please refer to that section for their explanations.

Table 6: Table of the methods used for dataset 2. The parameters of each method, the rationale behind the parameter choice, and the values the parameter takes are shown.

| Method | Parameter | Rationale | Values |
|---|---|---|---|
| SAX/MSAX | $w$ | factors of 360 | 2,3,4,5,6,8,9,10,12,15, 18,20,24,30,36,40,45, 60,72,90,120,180,360 |
| | $a$ | $2 \leq a \leq 25$, length of alphabet | $\overline{2, \ldots, 25}$ |
| HOT SAX/MSAX | $k$ | arbitrary | -1,25,50,75,100, 150,175,200,300 |
| | $m$ | same as $w$ | 2,3,4,5,6,8,9,10,12,15, 18,20,24,30,36,40,45, 60,72,90,120,180,360 |

This table provides values that create 4,968 parameter combinations when used with this research's programs to analyze the 48 ECGs of the MIT-BIH database. As a result, dataset 2 contains 238,464 files of ECGs analyzed with HOT SAX and HOT MSAX. Each of those combinations was applied to all 48 ECGs. As with dataset 1, these files were analyzed as stated in section 2.6. The mean values of the statistical measures for each set of unique parameters were calculated and the threshold of recall ≥95% applied. Table 7 shows the results of that analysis.

Table 7: Results of the analysis of dataset 2. The total number of parameter sets and the number and proportion of parameter sets fulfilling the conditions that recall be ≥95%.

| Method | Total Sets | Sets Satisfying recall ≥95% |
|---|---|---|
| S-SAX | 4,968 | 99 (1.2%) |
| D-SAX | | 192 (3.9%) |
| MSAX | | 255 (5.1%) |

As Table 7 shows, of the 4,968 total parameter sets, 99 have a recall of ≥95% for S-SAX, 192 for D-SAX, and 255 for MSAX. These dataset are large enough for the analysis to continue. As discussed in 2.6, the subsets of dataset 2 for which the recall is ≥95% will further be sorted by the precision, in descending order. Then, the top 10 values of S-SAX, D-SAX, and MSAX will be analyzed individually.

**Analysis of S-SAX** The top 10 values of S-SAX were first pruned by a threshold of recall ≥95% and then sorted descendingly by precision. Table 8 provides an overview of the parameters of these top 10 values, as well as their recall, accuracy, and precision.

Table 8: Table presenting a ranking of the top 10 most precise S-SAX parameter combinations and their parameters $k$, $w$, $m$, and $a$. The recall and accuracy values are also shown.

| Rank \ Properties | $k$ | $w$ | $m$ | $a$ | Recall (%) | Accuracy (%) | Precision (%) |
|---|---|---|---|---|---|---|---|
| 1 | -1 | 40 | 40 | 19 | 95.21 | 37.46 | 35.94 |
| 2 | -1 | 24 | 24 | 24 | 95.19 | 37.57 | 35.93 |
| 3 | -1 | 30 | 30 | 22 | 95.37 | 37.46 | 35.93 |
| 4 | -1 | 36 | 36 | 20 | 95.09 | 37.41 | 35.92 |
| 5 | -1 | 45 | 45 | 18 | 95.24 | 37.33 | 35.89 |
| 6 | -1 | 24 | 24 | 25 | 95.74 | 37.32 | 35.88 |
| 7 | -1 | 72 | 72 | 15 | 95.02 | 36.86 | 35.87 |
| 8 | -1 | 36 | 36 | 21 | 95.92 | 37.06 | 35.86 |
| 9 | -1 | 30 | 30 | 23 | 95.72 | 37.17 | 35.86 |
| 10 | -1 | 120 | 120 | 13 | 95.13 | 36.51 | 35.85 |

Table 8 shows that for the top 10 values by precision, the recall values are all approximately 95%, the accuracy is between 36% and 38%, and the precision is 36%. It is notable that all $k$ values are -1. To be able to choose the best parameter combination of these 10, their interquartile ranges and outliers for the recall value will be compared with the help of a boxplot. This boxplot is based on the set of 48 ECGs for each ranked method. This plot can be seen in Figure 7.

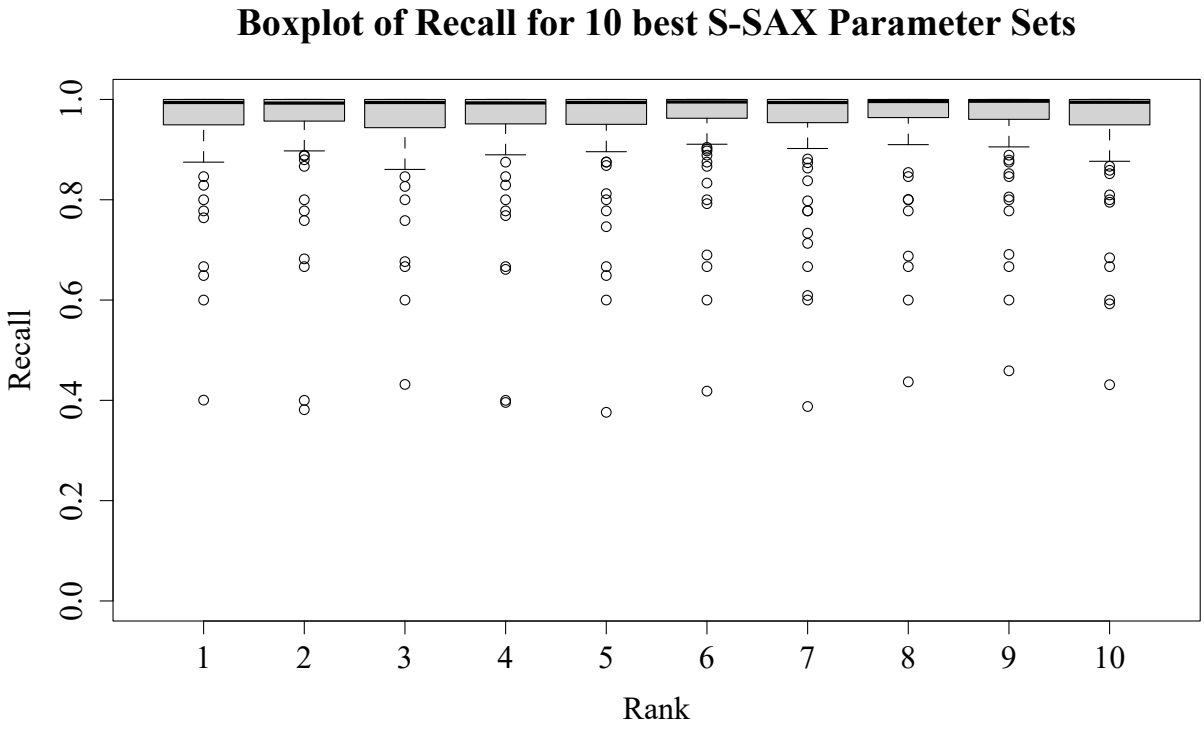


Figure 7: Boxplot showing the recall for the top 10 S-SAX parameter sets. The full list of parameters can be found in Table 8.

Figure 7 illustrates two important things. Firstly, there are only small differences between the recall values for any of the top 10 S-SAX parameter sets. Secondly, a clear pattern of 9 outliers

is visible.

To select the optimal parameter based on Figure 7, the interquartile range (IQR) is calculated for each rank. The use of IQR versus the mean is justified here because of the skewed data and the outliers. Ranks 6, 8, and 9 have the lowest IQR, at 0.0358, 0.0354, and 0.378, respectively. For further differentiation, the number of outliers is considered. Rank 6 has 13 outliers, rank 8 11, and rank 9 has 13. Considering these factors, rank 8 is chosen as the optimal parameter set for S-SAX. This set has the smallest IQR of the top 10 and the smallest number of outliers among those with small IQR values. With a $w$ value of 36, rank 8 represents a 10 time dimension reduction compared to the raw data.

**Analysis of D-SAX** For D-SAX, the top 10 values were also first pruned by a threshold of recall ≥95% and then sorted descendingly by precision. Table 9 provides an overview of the parameters of these top 10 values, as well as their recall, accuracy, and precision.

Table 9: Table presenting a ranking of the top 10 most precise D-SAX parameter combinations and their parameters $k$, $w$, $m$, and $a$. The recall and accuracy values are also shown.

| Properties / Rank | $k$ | $w$ | $m$ | $a$ | Recall (%) | Accuracy (%) | Precision (%) |
|---|---|---|---|---|---|---|---|
| 1 | -1 | 12 | 12 | 22 | 95.28 | 41.91 | 36.56 |
| 2 | -1 | 10 | 10 | 25 | 95.18 | 41.99 | 36.53 |
| 3 | -1 | 15 | 15 | 19 | 95.14 | 41.70 | 36.46 |
| 4 | -1 | 12 | 12 | 23 | 95.18 | 41.00 | 36.34 |
| 5 | -1 | 40 | 40 | 12 | 95.08 | 39.64 | 36.29 |
| 6 | -1 | 15 | 15 | 20 | 96.11 | 40.31 | 36.27 |
| 7 | -1 | 12 | 12 | 24 | 97.04 | 40.16 | 36.26 |
| 8 | -1 | 36 | 36 | 13 | 95.80 | 38.93 | 36.20 |
| 9 | -1 | 20 | 20 | 17 | 95.87 | 39.82 | 36.19 |
| 10 | -1 | 30 | 30 | 14 | 96.09 | 39.32 | 36.18 |

Table 9 shows that for the top 10 values by precision, like in the previous section. The recall values are between 95% and 97%, the accuracy between 39% and 42%, and the precision is 36%. Again, all $k = -1$. A boxplot comparing recall with rank is created in order to choose a best parameter combination. This plot is shown in Figure 8.

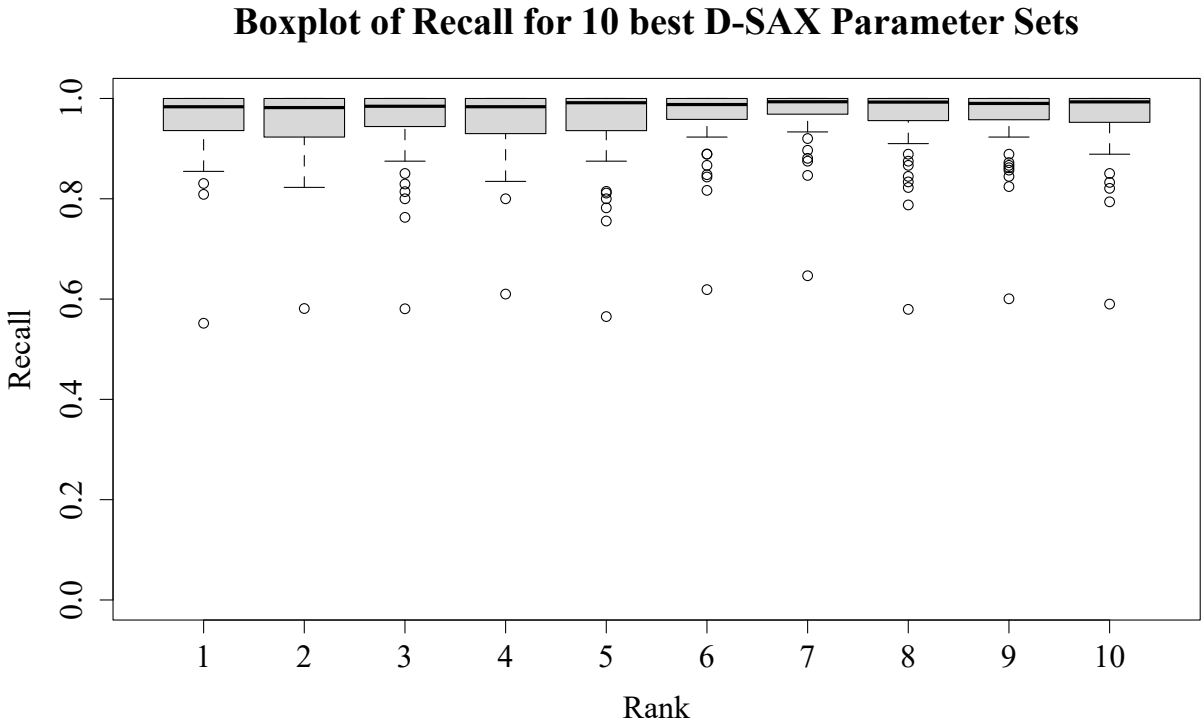


Figure 8: Boxplot showing the recall for the top 10 D-SAX parameter sets. The full list of parameters can be found in Table 9.

Visually, Figure 8 shows ranks 6, 7, and 9 to have the smallest IQR. This is confirmed by calculating the IQR. Rank 6 has an IQR of 0.038, rank 7 has 0.03, and rank 9 has 0.04. Regarding the number of outliers, rank 7 has 6, rank 6 has 7, and rank 9 has 8. Thus rank 7 is selected as the optimal parameter set. It has both the smallest IQR and the lowest number of outliers compared to other ranks with low IQRs. The dimension reduction ratio of rank 7 is 30.

**Analysis of MSAX** Lastly, the top 10 parameter sets of MSAX are considered. They, too, were selected based on recall ≥95% and sorted by precision. Table 10 shows the parameter set and statistical measures associated with the top 10 MSAX parameter combinations.

Table 10: Table presenting a ranking of the top 10 most precise MSAX parameter combinations and their parameters $k$, $w$, $m$, and $a$. The recall and accuracy values are also shown.

| Rank \ Properties | $k$ | $w$ | $m$ | $a$ | Recall (%) | Accuracy (%) | Precision (%) |
|---|---|---|---|---|---|---|---|
| 1 | -1 | 6 | 6 | 24 | 95.37 | 40.68 | 36.24 |
| 2 | -1 | 12 | 12 | 16 | 95.10 | 39.85 | 36.24 |
| 3 | -1 | 9 | 9 | 19 | 95.20 | 39.70 | 36.13 |
| 4 | -1 | 10 | 10 | 18 | 95.89 | 39.45 | 36.12 |
| 5 | -1 | 8 | 8 | 21 | 96.01 | 39.53 | 36.12 |
| 6 | -1 | 6 | 6 | 25 | 96.02 | 39.94 | 36.12 |
| 7 | -1 | 36 | 36 | 10 | 95.16 | 38.47 | 36.08 |
| 8 | -1 | 12 | 12 | 17 | 96.51 | 38.89 | 36.06 |
| 9 | -1 | 30 | 30 | 11 | 95.49 | 38.26 | 36.03 |
| 10 | -1 | 72 | 72 | 8 | 95.70 | 37.74 | 36.03 |

Table 10 shows, for the third time, that there is little difference between the statistical results of the top 10 best parameter sets. The recall values are all 95% to 96%, the accuracy is between 38% and 41%, and the precision is 36%. A boxplot is constructed for these 10 parameter sets to explore the interquartile range and outliers of MSAX. Figure 7 shows this boxplot.

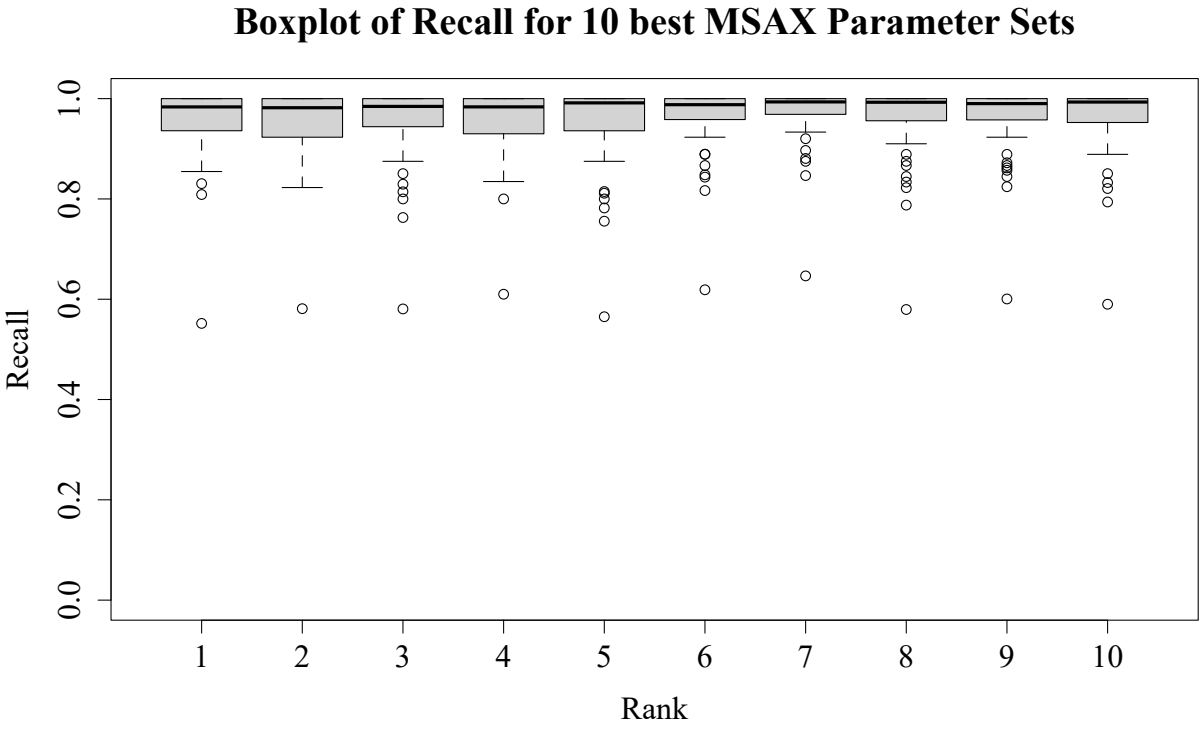


Figure 9: Boxplot showing the recall for the top 10 MSAX parameter sets. The full list of parameters can be found in Table 10.

In Figure 9, ranks 6, 8, and 10 visually stick out as having the lowest IQR. Calculating the IQR confirms that, yielding 0.038, 0.033, and 0.03 for ranks 6, 8, and 10 respectively. The number of outliers for rank 6 is 7, for rank 8 it is 6, and for rank 10 it is 8. Because rank 8 has the

second-lowest IQR and the smallest number of outliers, it is selected as the optimal parameter set for MSAX. Its $w$ value of 12 results in a dimension reduction of 30 compared to the raw data.

### 3.2.3 Comparison of Optimal Parameters

After determining the optimal parameter sets for the three methods investigated with the help of dataset 2, they will now be compared with each other. Table 11 summarizes the parameter sets that are optimal for S-SAX, D-SAX, and MSAX. When it comes to the method parameters, it is desirable for $w$ to be as small as possible to take advantage of the dimension reduction properties of SAX and MSAX. Accordingly, D-SAX and MSAX have the best $w$ parameters, as they are smaller than $w$ of S-SAX. Considering parameter $k$, they all have $k = -1$. The alphabet size $a$, while not influencing the dimension reduction, is relevant for the complexity of the method–lower values of $a$ result in a simpler representation. Accordingly, MSAX has the best value for $a$ because it is smaller than the competing methods.

Table 11: Table showing the optimal sets of parameters for S-SAX, D-SAX, and MSAX. Parameters $w$ and $m$ are combined because they are equal to each other. Best parameters are highlighted in bold.

| Method \ Parameter | $k$ | $w, m$ | $a$ |
|---|---|---|---|
| S-SAX | -1 | 36 | 21 |
| D-SAX | -1 | **12** | 24 |
| MSAX | -1 | **12** | **17** |

To further compare these sets of parameters, a boxplot of their recall values is created. Figure 10 shows this boxplot. Visually, the IQR of D-SAX look to be the smallest, S-SAX the largest. The outlier spread is also larger for S-SAX than for the other methods. The important variables regarding Figure 10 are summarized in Table 12.

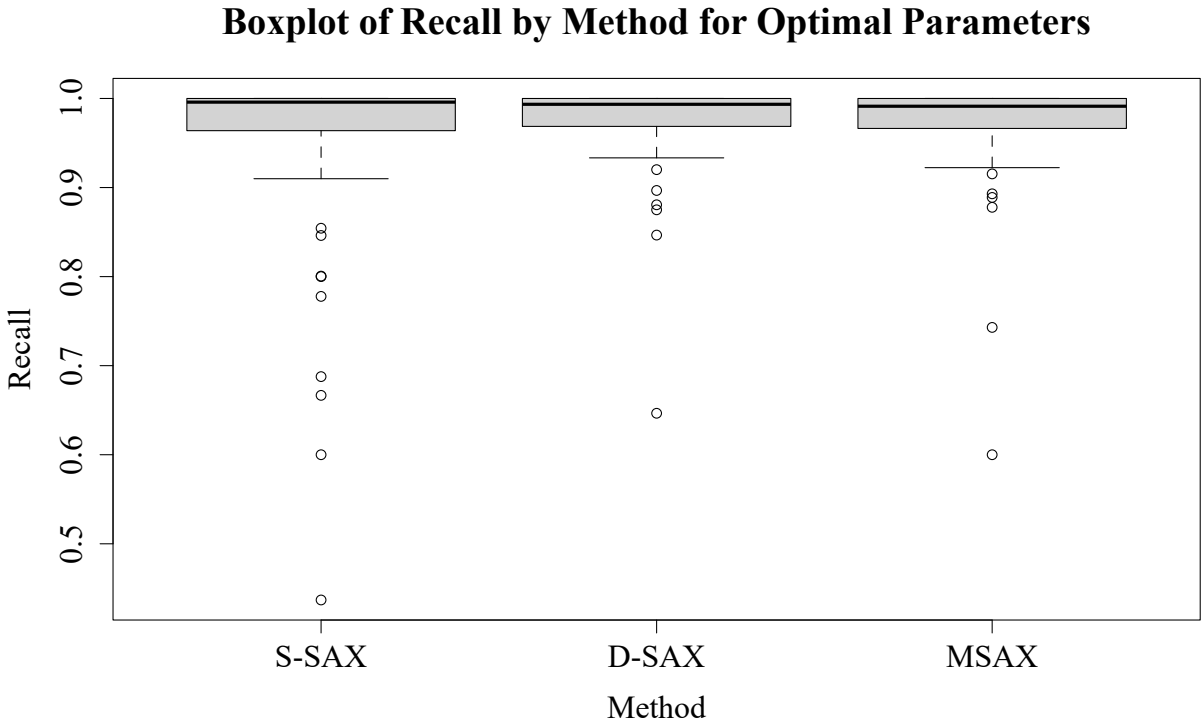


Figure 10: Boxplot showing the recall for the optimal parameter sets of each method. The full list of parameters can be found in Table 11.

Table 12: Table showing statistical measures for the recall of the optimal parameter sets for each method. Best values are highlighted in bold.

| Measure / Method | IQR | Median | Outliers |
|---|---|---|---|
| S-SAX | 0.035 | **99.60%** | 11 |
| D-SAX | **0.030** | 99.35% | **6** |
| MSAX | 0.033 | 99.13% | **6** |

Table 12 shows that D-SAX has the best IQR, S-SAX has the highest median recall, and both D-SAX and MSAX have the lowest number of outliers. As a final test, a possible biserial correlation between the methods and the resulting recall is investigated using R's standard `cor()` function, which calculates the Pearson correlation. The correlation coefficient of the recall depending on S-SAX or D-SAX is 0.06, for the recall depending on S-SAX or MSAX it is 0.033, and for the recall depending on D-SAX or MSAX it is -0.04. All three correlation coefficients are very close to zero, which indicates that there is no correlation between the methods used and the resulting recall value.

# 4 DISCUSSION

This section discussed the results presented in the previous section. It is this paper's goal to investigate how HOT MSAX influences the ECG discord discovery compared to HOT SAX. Additionally, the best parameters for high-recall discord discovery were sought. During the experiment phase, two datasets based on the MIT-BIH ECG database were used.

During the analysis of the first dataset, it was found that both HOT SAX and HOT MSAX perform poorly when the subsequence length parameter is not equal to the PAA segment count parameter. A possible cause of this is the way annotations were counted as True Positives and True Negatives. By only counting the annotations in the segment they fall into, a possible influence of the discord on the following segments is neglected. This could lead to HOT SAX and HOT MSAX identifying a discord caused by an abnormal heartbeat in the segment following the annotation and thus lead to the detection being labeled a False Positive. By making the subsequences as large as possible, the chance of this happening becomes smaller and accordingly, the recall value would increase. By setting the parameter $m$ equal to $w$ in the second dataset, the number and proportion of parameter sets satisfying the 95% recall condition increased.

When the second dataset was being analyzed, every set of parameters in the top 10 values of S-SAX, D-SAX, and MSAX had its $k = -1$. The parameter $k$ controls how many of the discords identified by HOT SAX and HOT MSAX will be returned after the computation is finished. Parameter $k$ being equal to -1 for the optimal parameters regardless of the method indicates that the highest concrete values of $k$, 300 for dataset 2, is too low to enable top recall values. This behavior may be explained by considering the precision values. If the average precision of HOT SAX and HOT MSAX is $\approx$ 36% and the goal is to maximize recall, restricting the number of detected discords that are used will directly lower the recall value.

For all parameter sets in dataset 2, comparing S-SAX, D-SAX, and MSAX showed that MSAX performs acceptably for 5.1% of all parameter combinations in dataset 2. S-SAX performed acceptably for 1.2% and D-SAX for 3.9%. From this is it inferred that MSAX is more robust when it comes to parameter choice. All three methods were tested with the same parameters, but MSAX performed acceptably for a larger share than the other methods. This is relevant because in applications involving unknown data, the parameter selection cannot be optimized without testing. A more robust method could still perform adequately in such a situation.

The optimal parameter sets (see Table 11) for the three methods all had a value of $k = -1$, which may be the result of the subsegment-annotation relationship mentioned above. The PAA segment values and thus the subsequence lengths for D-SAX and MSAX were 12 compared to 36 for S-SAX. The dimension reduction ratio for MSAX and D-SAX is 30, while it is 10 for S-SAX. A possible explanation for this is that MSAX and D-SAX are better suited to the multivariate discord detection tasks tested here. As a result of that, they do not require as much information as the less-effective S-SAX method to achieve the same recall values. The information requirement is relevant because, while PAA, SAX, and MSAX all lower-bound the Euclidean Distance and thus accurately represent the raw time series, each average trades a decrease in information about the

shape of the time series for a reduced dimension. If S-SAX is less effective at extracting information from a multivariate time series, higher numbers of PAA segments may be required to gain enough information to achieve high recall values.

The interquartile ranges of the recall value for all three optimal methods are not significantly different, all being close to 0.03. The median recall value is very similar across all three methods as well, roughly 99%. The number of outliers is 11 for S-SAX and 6 for the other two methods. The larger number of outliers for the S-SAX method could be attributed to it not considering the multivariate nature of ECGs. All observed outliers were outliers below the median recall value, indicating that certain ECGs are more difficult for the method to analyze than others. This is especially true for S-SAX, which does not properly take the nature of ECGs into account. This result should not be surprising, as both D-SAX and MSAX use all the information available, while S-SAX does not and is thus put at a disadvantage.

A correlation between the optimal S-SAX, D-SAX, or MSAX methods (Table 11) and the recall value has not been found. As a result, the hypothesis of this paper cannot be supported. There is no evidence of HOT MSAX improving the recall value compared to HOT SAX. Anacleto, Vinga, and Carvalho [9], in the paper introducing MSAX, analyzed a different ECG dataset as part of a collection of different time series. They used a $k$-Nearest-Neighbor classifier and found that there is no significant difference between the SAX representation and the MSAX representation when applied to ECG analysis. Their findings support this research's results.

# 5 CONCLUSION

This work has investigated the use of the MSAX time series representation for ECG discord discovery. This novel contribution is based on using the HOT SAX algorithm with the MSAX representation, a process called HOT MSAX in this research. The MIT-BIH ECG database was used to evaluate the HOT SAX and HOT MSAX algorithms experimentally. Each algorithm's parameters were optimized based on the results of the evaluation. The recall value for HOT MSAX was compared to the recall value for HOT SAX when both are using optimal parameters. A correlation test was performed to determine if HOT MSAX increases the recall value compared to HOT SAX. No correlation was found between the methods used with optimal parameters and the resulting recall. Accordingly, this work's hypothesis cannot be supported because HOT MSAX did not significantly improve discord discovery recall compared to HOT SAX.

The application of the MSAX representation to ECG analysis is novel and follows the future applications mentioned in by Anacleto, Vinga, and Carvalho [9], the creators of the MSAX method. A further contribution to the literature is the application of the HOT SAX algorithm to the MSAX representation. This expands the use of HOT SAX to multivariate time series in the form of HOT MSAX. In possible future research, HOT MSAX can be applied to different and unknown datasets to test the optimality of the parameters determined in this work and judge its performance in discovering different types of ECG discords, for example those caused by ischaemic heart disease.

# REFERENCES


[1] *The top 10 causes of death*, 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (visited on 05/25/2021).

[2] Institute of Medicine (US) Committee on Social Security Cardiovascular Disability Criteria, "Ischemic Heart Disease," 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).

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

[4] P. Kligfield, L. S. Gettes, J. J. Bailey, *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*," *Circulation*, vol. 115, no. 10, pp. 1306–1324, 2007. DOI: 10.1161/CIRCULATIONAHA.106.180200.

[5] L. Xie, Z. Li, Y. Zhou, *et al.*, "Computational Diagnostic Techniques for Electrocardiogram Signal Analysis," *Sensors*, vol. 20, no. 21, p. 6318, 2020. DOI: 10.3390/s20216318.

[6] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," 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.

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

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

[9] M. Anacleto, S. Vinga, and A. M. Carvalho, "MSAX: Multivariate Symbolic Aggregate Approximation for Time Series Classification," 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. DOI: 10.1007/978-3-030-63061-4_9.

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

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

[12] G. Moody and R. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. DOI: 10.1109/51.932724.

[13] A. L. Goldberger, L. A. N. Amaral, L. Glass, *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, 2000. DOI: 10.1161/01.CIR.101.23.e215.

[14] J. Shieh and E. Keogh, "*I* SAX: Indexing and mining terabyte sized time series," 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. DOI: 10.1145/1401890.1401966.

[15] C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A Novel Bit Level Time Series Representation with Implication of Similarity Search and Clustering," 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. DOI: 10.1007/11430919_90.

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

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

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

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

[20] R. Nygaard and D. Haugland, "Compressing ECG signals by piecewise polynomial approximation," 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. DOI: 10.1109/ICASSP.1998.681812.

[21] 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," *PLOS ONE*, vol. 13, no. 10, J. Zhao, Ed., e0206170, 2018. DOI: 10.1371/journal.pone.0206170.

[22] 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, 2016, pp. 72–80. DOI: 10.1145/3011141.3011146.

[23] Y. Sun, J. Li, J. Liu, *et al.*, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189–198, 2014. DOI: 10.1016/j.neucom.2014.01.045.

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

[25] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: Extension of Symbolic Aggregate Approximation for Financial Time Series Data Representation," 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).

[26] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, "1d-SAX: A Novel Symbolic Representation for Time Series," 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. DOI: 10.1007/978-3-642-41398-8_24.

[27] M. M. M. Fuad and P.-F. Marteau, "TOWARDS A FASTER SYMBOLIC AGGREGATE APPROXIMATION METHOD:" 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. DOI: 10.5220/0003006703050310.

[28] 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*, 2010, pp. 58–67. DOI: 10.1109/ICDM.2010.124.

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

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

[31] C. Zhang, Y. Chen, A. Yin, *et al.*, "Anomaly detection in ECG based on trend symbolic aggregate approximation," *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 2154–2167, 2019. DOI: 10.3934/mbe.2019105.

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

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

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

[35] A. H. Kadish, A. E. Buxton, H. L. Kennedy, *et al.*, "ACC/AHA Clinical Competence Statement on Electrocardiography and Ambulatory Electrocardiography," *Circulation*, vol. 104, no. 25, pp. 3169–3178, 2001. DOI: 10.1161/circ.104.25.3169.

[36] C. Antzelevitch and A. Burashnikov, "Overview of Basic Mechanisms of Cardiac Arrhythmia," *Cardiac electrophysiology clinics*, vol. 3, no. 1, pp. 23–45, 2011. DOI: 10.1016/j.ccep.2010.10.012.

[37] A. N. Nowbar, M. Gitto, J. P. Howard, *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," *Circulation: Cardiovascular Quality and Outcomes*, vol. 12, no. 6, 2019. DOI: 10.1161/CIRCOUTCOMES.118.005375.

[38] N. Herring, "ECG diagnosis of acute ischaemia and infarction: Past, present and future," *QJM*, vol. 99, no. 4, pp. 219–230, 2006. DOI: 10.1093/qjmed/hcl025.

[39] H. Sivaraks and C. A. Ratanamahatana, "Robust and Accurate Anomaly Detection in ECG Artifacts Using Time Series Motif Discovery," *Computational and Mathematical Methods in Medicine*, vol. 2015, pp. 1–20, 2015. DOI: 10.1155/2015/453214.

[40] R. Valupadasu and B. R. R. Chunduri, "Identification of Cardiac Ischemia Using Spectral Domain Analysis of Electrocardiogram," in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom: IEEE, 2012, pp. 92–96. DOI: 10.1109/UKSim.2012.22.

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

[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," in *Computers in Cardiology 1997*, Lund, Sweden: IEEE, 1997, pp. 673–676. DOI: 10.1109/CIC.1997.648140.

[43] A. Taddei, G. Distante, M. Emdin, *et al.*, "The European ST-T database: Standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography," *European Heart Journal*, vol. 13, no. 9, pp. 1164–1172, 1992. DOI: 10.1093/oxfordjournals.eurheartj.a060332.

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

[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," *Advanced Engineering Informatics*, vol. 40, pp. 121–134, 2019. DOI: 10.1016/j.aei.2019.03.003.

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

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

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

[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)*, 2018, pp. 2707–2710. DOI: 10.1109/EMBC.2018.8512761.

[50] N. D. Pham, Q. L. Le, and T. K. Dang, "HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery," 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. DOI: 10.1007/978-3-642-12145-6_12.

[51] H. Tayebi, S. Krishnaswamy, A. B. Waluyo, *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, 2011, pp. 289–290. DOI: 10.1109/MDM.2011.67.