1 METHODS

TODO fix sectioning and verbal toc

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

While MSAX and SAX both are time series representation methods, they can be applied to ECGs, as EGCs are discrete multivariate time series. Mathematically, a discrete time series is a series of T observations made at discrete points in time, with n values recorded at each moment in time. Following [1], anacleto2020

$$\{\mathbf{v}[t]\}_{t\in\{1,\dots,T\}} \tag{1.1}$$ 08:eq

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

$$\mathbf{v}[t] = (v_1[t], \dots, v_n[t])^T \tag{1.2}$$ 08:eq

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 (1.2). 08:eq:ts-point As virtually all ECGs consist of more than one lead ($n > 1$), ECGs are multivariate time series.

1.0.1 SAX

The Symbolic Aggregate Approximation, introduced in 2003 by Lin, Keogh, Lonardi, and Chiu, is a symbolic time series representation [2]. lin2003 lin2003 lin2003 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 [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] and application to ECGs [16], zhang2019 makes SAX a promising methods to use. The SAX representation only works for time series $\mathbf{v}[t]$ for which $n = 1$, i.e. which are univariate. Thus (1.1) 08:eq:ts becomes $\mathbf{v}[t] = v_1[t]$. Using the SAX representation is a three-step process. Firstly,

the raw time series is normalized. Secondly, the dimension of the normalized time series is reduced using PAA. Thirdly, the PAA-represented time series is discretized. Additionally, a distance measure between two SAX representations is defined.

Normalization. The normalization for the SAX representation is necessary because, to compare two time series, it is standard practice to normalize both of them because otherwise comparisons between them are not useful [lin2003]. 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.

TODO insert figure here

08:paa-proc

Dimension reduction with PAA. The dimension reduction of the SAX representation is due to the use of PAA. The PAA method takes a univariate time series $\mathbf{v}[t]$ of length T and an integer w and segments $\mathbf{v}[t]$ into w segments, taking the average of each. Following [lin2003], 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 [lin2003]

$$\bar{\mathbf{v}}[t] = \frac{w}{T} \sum_{j=\frac{n}{w}(t-1)+1}^{\frac{n}{w}t} \mathbf{v}[t], \qquad \forall t \in \{1, \dots, w\}. \tag{1.3}$$

08:eq

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.

TODO insert figure here

Table 1.1: Breakpoint values for numbers of breakpoints a from 3 to 6. The parameter a determines into how many equally-sized areas the normal curve $\mathcal{N}(0,1)$ is split. The breakpoints β_i delimit the areas. Table contents are quoted from [2]. lin2003

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

Discretization of PAA representation. This last step in the SAX representation process involves transforming the PAA representation $\bar{\mathbf{v}}[t]$ 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, lin2003 Lin, Keogh, Lonardi, and Chiu [2] use so-called "breakpoints". These breakpoints are a sorted list of numbers $B = \beta_1, \ldots, \beta_{a-1}$. The area under the normal curve $\mathcal{N}(0,1)$ (i.e. the probability) between two consecutive segments β_i and $\beta_{i+1} = 1/a$. This creates a segments ($a-1$ breakpoints) of $\mathcal{N}(0,1)$ that have the same area, i.e. the same probability. The values of the breakpoints in B can be found in a Z-table. For illustration, Table 1.1 shows the breakpoint values for $a = 3$ to $a = 6$. 08:tab:B

Once the breakpoint values have been determined, the discretization process begins. The process assigns all PAA segments whose value is below β_1 the symbol "a". The PAA segments falling in the area $\beta_1 \leq$ and $< \beta_2$ are assigned "b". This mapping process is continued, until all PAA segments are symbolized. Now we have arrived at the SAX representation. The SAX representation of $\bar{\mathbf{v}}[t]$ is denoted $\hat{\mathbf{v}}[t]$ and has the same length as $\bar{\mathbf{v}}[t]$ (w). Mathematically, the discretization process is formulated in [2] as lin2003

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

Here alpha_j is the jth letter of the alphabet, i.e. alpha_1 = "a", alpha_2 = "b" … The resulting time series representation has an even more reduced dimension than PAA because instead of infinitely many possible values for the real-valued PAA values, now there are only a different, equiprobable symbols. Thus, the SAX representation $\hat{\mathbf{v}}[t]$ has been obtained. TODO insert graph here

SAX distance measure. A distance measure between two SAX representations of the same length is required to be able to compare them with each other. The SAX distance function is based on the Euclidean Distance between two time series $\mathbf{v}[t]$ and $\mathbf{u}[t]$ is [2] lin2003

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

Table 1.2: 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 (1.5). 08:eq:dist-cell

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

Through the PAA distance as an intermediate step, the authors arrive at MINDIST in (1.4), the SAX distance function that returns the minimum distance between the two original time series. It is defined as [2] lin2003

$$\text{MINDIST}\left(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t]\right) \equiv \sqrt{\frac{T}{w}} \sqrt{\sum_{t=1}^{w} \left(\text{dist}(\hat{\mathbf{u}}[t], \hat{\mathbf{v}}[t])\right)^2}. \tag{1.4}$$

The function dist is based on a lookup table that contains the distances between two symbols. Table 1.2 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

$$\text{cell}_{r,c} = \begin{cases} 0, & \text{if } |r - c| \leq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & \text{otherwise} \end{cases} \tag{1.5}$$

is used to calculate the values of each cell in Table 1.2 by r (row) and c (column) [2]. TODO inset example of distance between a short segment of SAX stuff, maybe two ecg segments

1.0.2 MSAX

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

Normalization. The rationale for normalization in the MSAX representation is twofold. Firstly, the same considerations as for SAX apply with regards to comparing two time series. Secondly, MSAX utilizes multivariate normalization to take advantage of the covariance structure of multivariate time series data. To avoid confusion with the previous section, a multivariate time series shall be denoted as $\mathbf{V}[t]$. Multivariate normalization relies on a sample mean vector containing the sample mean for each of the time series $(V_1[t], \dots, V_n[t])^T$ 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 [1]: anacleto2020

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

The covariance matrix, an $n \times n$ matrix, contains the variance of each part $(V_1[t], \dots, V_n[t])^T$ of $\mathbf{V}[t]$ on its main diagonal, and the covariance between ith and jth parts of $\mathbf{V}[t]$ in the (i, j) position. The general form of a covariance matrix is shown in (1.6) below. The covariance matrix is denoted as $\text{Var}(\mathbf{V}[t])$ or $\Sigma_{n\times n}$. It is calculated as: 08:eq:cov-mat

$$\text{Var}(\mathbf{V}[t]) = \Sigma_{n\times n} = \begin{bmatrix} \text{cov}(V_1, V_1) & \dots & \text{cov}(V_1, V_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(V_n, V_1) & \dots & \text{cov}(V_n, V_n) \end{bmatrix}. \tag{1.6}$$

08:eq

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

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

(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 [1]: anacleto2020

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

The result will have a mean of zero and uncorrelated variables [1]. anacleto2020

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

Dimension reduction with PAA. Dimension reduction using PAA for MSAX is performed in exactly the same way as for SAX. The procedure outlined in the previous section is applied to each of the elements $(V_1[t], \dots, V_n[t])^T$ of the time series $\mathbf{V}[t]$–equation (1.3) is applied to each part. This results in a PAA representation of the original time series $\overline{\mathbf{V}}[t] = \left(\overline{V_1}[t], \dots, \overline{V_n}[t]\right)^T$. 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 [1] 08:eq:paa anacleto2020 TODO insert figure here

Discretization of PAA representation. The discretization of the PAA representation for MSAX also works like it does for SAX. Like in the previous paragraph, the process used in the SAX representation is applied to each of the sub-time series in $\overline{\mathbf{V}}[t]$ to obtain $\widehat{\mathbf{V}}[t]$. The alphabet size a is the same for each $V_n[t]$ and the symbols are found in the same way as in the SAX representation. The breakpoint values are calculated the same and Table 1.1 is as valid for MSAX as it is for SAX. The assigning of symbols is generally performed in the same way, too. For bivariate time series ($n = 2$), $V_1[t]$ could be assigned lowercase symbols ("a", "b" …) while $V_2[t]$ could be assigned uppercase symbols ("A", "B" …). This has no impact on the method, it is simply a visual aid for the viewer to distinguish the values. The final MSAX representation $\widehat{\mathbf{V}}[t]$ will consist of one long list of symbols because for each moment t all generated symbols are combined into a list for this time that represent all sub-time series at that time. TODO add a graph and example here for illustration

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

$$\text{MINDIST\_MSAX}\left(\widehat{\mathbf{U}}[t], \widehat{\mathbf{V}}[t]\right) \equiv \sqrt{\frac{T}{w}} \sqrt{\sum_{t=1}^{w} \left(\sum_{i=1}^{n} \left(\text{dist}(\widehat{U}_i[t], \widehat{V}_i[t])\right)^2 \right)}. \tag{1.7}$$

The function dist is the same as the SAX function, being based on equation (1.5) and lookup tables like Table 1.2. Like MINDIST, MINDIST_MSAX also lower bounds the Euclidean Distance and derives all the same benefits from that. TODO inset example of distance between a short segment of SAX stuff, maybe two ecg segments

1.0.3 HOT SAX

The Heuristically Ordered Time series using Symbolic Aggregate Approximation is a discord discovery algorithm introduced by Keogh, Lin, and Fu in 2005 [15]. 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 [15]. 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 [15]. As each of these loops would iterate over all of the m subsegments, resulting in the mentioned m^2 complexity.

HOT SAX has the goal of speeding up this process and making discord discovery viable

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

The magic heuristic can of course not exist, hence the name. But keogh2005 Keogh, Lin, and Fu approximate it to still achieve a significant speedup. The first step the authors take is to apply the SAX representation to the time series to reduce its complexity and dimension, while retaining an accurate representation of the data [lin2003 2]. To compare two subsegments, the SAX distance function MINDIST is used. Then, a certain window size is chosen that represents the number of SAX segments that will make up one of the aforementioned time series subsegments. Now the magic heuristic can be approximated. The outer heuristic, which returns the subsegments of the time series in descending order by their distance to their neighbors. The authors approximate this by taking each SAX subsegment and insert them into an array, counting how many times each unique time series occurs. By sorting this array by the occurrence counts, the outer heuristic can be approximated. At the same time as the outer heuristic, the inner heuristic can be approximated. For its approximation, a digital tree (also known as trie or prefix tree) is used. In this tree, the SAX representation is used as an index to locate a leaf node. This node contains the locations in the time series where the particular subsequence occurs. Effectively, this prefix tree can be used to locate all SAX subsequences that are identical to a given one. So, if one as the subsegment "abc", it is possible to find all other locations of the "abc" subsegment in the time series using this tree. TODO use Figure 4 from [keogh2005] here By default, HOT SAX only returns the most discordant time series subsegment. Because more than one time series discord can be present in a time series (as in an ECG), it is useful to extract more than one. The number of discords to be extracted is called k. If $k > 1$, each newly found discord is compared to the other already found discords, and only the k discords with the largest distance values are saved. In this way, it is possible to extract an exact number of discords.

Another modification is to save all discords that the method detects [15].

The authors find that HOT SAX can effectively detect time series discords and speed up the process when compared to the brute force method. A strength of the method is, in their opinion, the fact that it only requires the user to determine a single parameter, the length of the subsequence (the parameter k has no influence on the method itself, it simply determines how many of the results should be used) [15]. 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 [15]. For these reasons, HOT SAX was chosen as this research's discord discovery method. While HOT SAX was designed to work with SAX, all it requires is a time series representation and a lower-bounding distance measure defined on it. MSAX exhibits both of those traits and can thus be used with HOT SAX. Doing this expands HOT SAX to multivariate time series, as MSAX can represent those. Using HOT SAX with the MSAX representation is novel and a contribution of this research.

2 RESULTS

TODO fix up this section, set proper headers, etc

2.1 Implementation

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

All code used in the implementation of these methods is available upon request via email at konarski_m@auca.kg

.

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

2.1.1 The ECG Data

The ECG data used in this research is the MIT-BIH Arrhythmia Database [17, 18]. This database contains 48 ECG recordings that are each 30 minutes long. For each of the ECGs, this database contains two ECG leads and is thus multivariate data. The leads chosen are not the same in each ECG, they were chosen based on which of the 12 originally recorded ones best represent the condition of the ECG. A team of cardiologists annotated each heartbeat in each ECG and determined if it is a normal heartbeat or not. For example, a beat with the annotation "N" is a normal beat, while "A" denotes an atrial premature beat. The annotations also include non-beat features such as a change in signal quality, denoted by "~", or a rhythm change, which is denoted as "+". A full list of annotations and their meaning is available at https://archive.physionet.org/physiobank/annotations.shtml. These annotations make it possible to judge the performance of a discord detection algorithm, as each detected discord can be checked for correctness using the provided annotations. While the MIT-BIH database is not the only database that possesses such annotations, it is one of the most commonly used ones in the literature (see [16, 19, 20, 21, 22, 23, 24]) and it represents a middle ground in a couple important respects. The databases 48 ECGs are a manageable number, falling in between the extremes of around 10 and over 100 ECGs. Furthermore, the 30 minutes length represent real-world ECGs better than 10 second excerpts, but are not as long and analysis-intensive as 24 hour recordings. Lastly, the MIT-BIH database has a sampling frequency of 360 samples per second, which is an adequate value [25]. The ECG data in this database is unfiltered.

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

`.hea`

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

`.dat`

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

`rdsamp`

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

`rdann`

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

2.1.2 SAX, MSAX, HOT SAX Implementation

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

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

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

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

Table 2.1: 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

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

2.1.3 Statistical Analysis of Results

After completing the computations for different sets of parameters, the results need to be analyzed. While HOT SAX is not a classifier in the sense of classifying heartbeats by medical standards, it does classify them into discords and non-discords. Thus, it is a binary classifier. Binary classifiers can be evaluated using the well-known True Positive, True Negative, False Negative, and False Positive values. Table 2.1 shows their relationship. The values in Table 2.1 can be used to calculate many useful ratios that assist the evaluation of the HOT SAX algorithm. This research uses the recall value (also known as sensitivity), the accuracy, and the precision. These ratios are calculated as follows: recall value is defined as

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

the accuracy as

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

and the precision as

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

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

The analysis of the methods was performed using the data whose generation was discussed above. The analysis was performed using the R programming language. R is an established statistical and mathematical programming language with great support for statistical methods and tests. The first step in the analysis was the processing of the data generated using the Julia program. This consisted of calculating the True Positive, True Negative, False Negative, and False Positive values for each parameter combination and each method. A segment was considered a “non-discord” if its annotation consisted of an “N” or nothing “”. The former is obvious; the decision to consider no annotation (“”) a non-discord was made because for certain segments of the ECGs, no annotations were available. This can happen when the subsequence length for HOT SAX

All files were analyzed with respect to the performance of MSAX.

To analyze the performance of HOT SAX with SAX and MSAX as representations,

- explain true positive, true negative, and so on
- explain recall, accuracy, precision, f1
- explain why recall was chosen and if that is fair
- introduce the correlations that we would expect to find if my hypothesis is true and also the ones that would disprove it
- which types of correlation, significance testing, and modeling will be used and why; what are the justifications

TODO the idea is to use the ECG as it would be recorded or digitized by anyone, without filtering. see zhang2019 if they did filtering

TODO make a final applications section that explains how HOT SAX will be used with each of the time series

TODO why can filtering be ignored? put this as further research to investigate the influence of filtering on this process

TODO give good reasons why I chose the methods and data bases

TODO goals:
- reader can assess believability of results
- all information necessary to replicate the research
- describe all materials, procedure, ect
- all the formulae
- state all the limitations of the methods and the ones I impose myself
- analytical methods and languages

TODO answer questions:
- can someone else accurately replicate the study
- can the data be obtained again
- are all parts / instruments described with enough accuracy
- is the data freely available
- can the statistical analysis be repeated
- can the algorithms be replicated?

TODO Sections:
- general overview -> flowchart
- then explain each element of the flowchart one by one
- use formulae etc
- nice amount of tikz graphs
- section on implementation with details and the more important elements
use another flow chart?
- use graphs to illustrate all important elements
- make a data description section that describes my process of data handling; which database
- explain the parameters that the methods have and what they mean
- describe how a got all the data

This section explains the methods used in this research. TODO create flow charts for all this shit to make it simpler. First methods section for the analytical methods in a mathematical way.

2.2 Statistical Evaluation

- reading the data into R
- summarizing the data
- the summarized data files
- libraries used

TODO must include:
- present the actual results and findings

- range of validity
- DO NOT INTERPRET
- mention positive and negative results
- give enough information for others to make their own judgements
- use subheadings
- key results in clear statements at paragraph beginnings
- could be short
-

TODO Sections

- use confusion matrices for what is is vs what was predicted [p. 44 anacleto2019]
- compare all the parameters and their influence
-

2.3 First Run

- parameters for this run
- why could I not let it continue
- what did this run indicate -> what did I change and modify for the next run
- keep in mind that the lower recall can be caused by the way I do the ECG checking, and that I did not want to assign data to segments that did not have it before out of fear that I would invent results.

2.4 Second Run

2.5 SAX

influence and significance of all the major parameters:

- k
- paa count
- subsequence count
- alphabet size
- which ones seem to be the best

2.6 MSAX

influence and significance of all the major parameters:

- k
- paa count
- subsequence count
- alphabet size
- which ones seem to be the best

2.7 MSAX vs SAX

Comparing SAX to MSAX is done using the recall value defined in TODO reference. Investigating the correlation between the methods (represented by a 1 for SAX and a 0 for MSAX), yields the correlation coefficient of -0.25. This coefficient indicates that for all investigated parameter combinations, the use of the MSAX method is weakly correlated with an increase in recall. When a specific set of parameters is selected and the correlation analysis is repeated, the correlation coefficient is -0.73, indicating a strong correlation. Here $k = -1$ and paa_count = 12.

- just the results that are gained directly from the data
- put results in graphs and tables to make them referencable

REFERENCES

anacleto2020 [1] M. Anacleto, S. Vinga, and A. M. Carvalho, "MSAX: Multivariate Symbolic Aggregate Approximation for Time Series Classification," en, in *Computational Intelligence Methods for Bioinformatics and Biostatistics*, P. Cazzaniga, D. Besozzi, I. Merelli, and L. Manzoni, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 90–97, ISBN: 978-3-030-63061-4. DOI: 10.1007/978-3-030-63061-4_9.

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

aremu2019 [3] O. O. Aremu, D. Hyland-Wood, and P. R. McAree, "A Relative Entropy Weibull-SAX framework for health indices construction and health stage division in degradation modeling of multivariate time series asset data," en, *Advanced Engineering Informatics*, vol. 40, pp. 121–134, Apr. 2019, ISSN: 1474-0346. DOI: 10.1016/j.aei.2019.03.003.

fuad2010 [4] M. M. M. Fuad and P.-F. Marteau, "TOWARDS A FASTER SYMBOLIC AGGREGATE APPROXIMATION METHOD:" en, in *Proceedings of the 5th International Conference on Software and Data Technologies*, University of Piraeus, Greece: SciTePress - Science and and Technology Publications, 2010, pp. 305–310, ISBN: 978-989-8425-22-5 978-989-8425-23-2. DOI: 10.5220/0003006703050310.

guigou2017 [5] F. Guigou, P. Collet, and P. Parrend, *Anomaly Detection and Motif Discovery in Symbolic Representations of Time Series*. Apr. 2017. DOI: 10.13140/RG.2.2.20158.69447.

he2020 [6] Z. He, S. Long, X. Ma, and H. Zhao, "A Boundary Distance-Based Symbolic Aggregate Approximation Method for Time Series Data," en, *Algorithms*, vol. 13, no. 11, p. 284, Nov. 2020. DOI: 10.3390/a13110284.

lahcioglu2021 [7] B. Kulahcioglu, S. Ozdemir, and B. Kumova, "Application of Symbolic Piecewise Aggregate Approximation (PAA) Analysis to ECG Signals," Mar. 2021.

liu2018 [8] M. Liu and Y. Kim, "Classification of Heart Diseases Based On ECG Signals Using Long Short-Term Memory," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2018, pp. 2707–2710. DOI: 10.1109/EMBC.2018.8512761.

lkhagva2006 [9] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: Extension of Symbolic Aggregate Approximation for Financial Time Series Data Representation," en, in *Proceeding of IEICE the 17th Data Engineering Workshop*, Ginowan, Japan, 2006, p. 7. [Online]. Available: https://www.researchgate.net/publication/229046404_Extended_SAX_extension_of_symbolic_aggregate_approximation_for_financial_time_series_data_representation (visited on 02/27/2021).

alinowski2013 [10] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, "1d-SAX: A Novel Symbolic Representation for Time Series," en, in *Advances in Intelligent Data Analysis XII*, A. Tucker, F. Höppner, A. Siebes, and S. Swift, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 273–284, ISBN: 978-3-642-41398-8. DOI: 10.1007/978-3-642-41398-8_24.

ordonez2008 [11] P. Ordóñez *et al.*, "Visualizing Multivariate Time Series Data to Detect Specific Medical Conditions," *AMIA Annual Symposium Proceedings*, vol. 2008, pp. 530–534, 2008, ISSN: 1942-597X. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2656052/ (visited on 03/30/2021).

pham2010 [12] N. D. Pham, Q. L. Le, and T. K. Dang, "HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery," en, in *Intelligent Information and Database Systems*, N. T. Nguyen, M. T. Le, and J. Świątek, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2010, pp. 113–121, ISBN: 978-3-642-12145-6. DOI: 10.1007/978-3-642-12145-6_12.

tayebi2011 [13] H. Tayebi *et al.*, "RA-SAX: Resource-Aware Symbolic Aggregate Approximation for Mobile ECG Analysis," in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, Jun. 2011, pp. 289–290. DOI: 10.1109/MDM.2011.67.

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

keogh2005 [15] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence," en, in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, TX, USA: IEEE, 2005, pp. 226–233, ISBN: 978-0-7695-2278-4. DOI: 10.1109/ICDM.2005.79.

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

moody2001 [17] G. Moody and R. Mark, "The impact of the MIT-BIH Arrhythmia Database," en, *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May-June/2001, ISSN: 07395175. DOI: 10.1109/51.932724.

oldberger2000 [18] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," en, *Circulation*, vol. 101, no. 23, Jun. 2000, ISSN: 0009-7322, 1524-4539. DOI: 10.1161/01.CIR.101.23.e215.

prasad2018 [19] B. V. P. Prasad and V. Parthasarathy, "Detection and classification of cardiovascular abnormalities using FFT based multi-objective genetic algorithm," en, *Biotechnology & Biotechnological Equipment*, vol. 32, no. 1, pp. 183–193, Jan. 2018, ISSN: 1310-2818, 1314-3530. DOI: 10.1080/13102818.2017.1389303.

kanani2020 [20] P. Kanani and M. Padole, "ECG Heartbeat Arrhythmia Classification Using Time-Series Augmented Signals and Deep Learning Approach," en, *Procedia Computer Science*, Third International Conference on Computing and Network Communications (CoCoNet'19), vol. 171, pp. 524–531, Jan. 2020, ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.04.056.

kaur2016 [21] I. Kaur, R. Rajni, and A. Marwaha, "ECG Signal Analysis and Arrhythmia Detection using Wavelet Transform," en, *Journal of The Institution of Engineers (India): Series B*, vol. 97, no. 4, pp. 499–507, Dec. 2016, ISSN: 2250-2106, 2250-2114. DOI: 10.1007/s40031-016-0247-3.

nygaard1998 [22] R. Nygaard and D. Haugland, "Compressing ECG signals by piecewise polynomial approximation," en, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 3, Seattle, WA, USA: IEEE, 1998, pp. 1809–1812, ISBN: 978-0-7803-4428-0. DOI: 10.1109/ICASSP.1998.681812.

sivaraks2015 [23] H. Sivaraks and C. A. Ratanamahatana, "Robust and Accurate Anomaly Detection in ECG Artifacts Using Time Series Motif Discovery," en, *Computational and Mathematical Methods in Medicine*, vol. 2015, pp. 1–20, 2015, ISSN: 1748-670X, 1748-6718. DOI: 10.1155/2015/453214.

alupadasu2012 [24] R. Valupadasu and B. R. R. Chunduri, "Identification of Cardiac Ischemia Using Spectral Domain Analysis of Electrocardiogram," en, in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom: IEEE, Mar. 2012, pp. 92–96, ISBN: 978-1-4673-1366-7 978-0-7695-4682-7. DOI: 10.1109/UKSim.2012.22.

kligfield2007 [25] P. Kligfield *et al.*, "Recommendations for the Standardization and Interpretation of the Electrocardiogram: Part I: The Electrocardiogram and Its Technology: A Scientific Statement From the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society *Endorsed by the International Society for Computerized Electrocardiology*," en, *Circulation*, vol. 115, no. 10, pp. 1306–1324, Mar. 2007, ISSN: 0009-7322, 1524-4539. DOI: 10.1161/CIRCULATIONAHA.106.180200.