DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ANOMALY DETECTION IN TEMPORAL DATA MINING

by

Mehmet Yavuz ONAT

July, 2015

İZMİR

ANOMALY DETECTION IN TEMPORAL DATA MINING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of Science in
Statistics, Statistics Program

by

Mehmet Yavuz ONAT

July, 2015
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"ANOMALY DETECTION IN TEMPORAL DATA MINING"** completed by **MEHMET YAVUZ ONAT** under supervision of **ASST. PROF. ENGİN YILDIZTEPE** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Engin YILDIZTEPE

Supervisor

Prof. Dr. E. Erkuzan

(Jury Member)

Prof. Dr. Celal Çelebi

(Jury Member)

Prof.Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Asst. Prof. Engin YILDIZTEPE, for his excellent guidance and providing me with a perfect atmosphere to do my research. Besides my thesis I also would like to thank to him sharing his experiences and for the new perspectives in life that I gained during my master's degree programme.

I would like to acknowledge my dissertation committee members, Prof. Dr. Esin FİRUZAN and Prof. Dr. Cenk ÖZLER, for their constructive comments and suggestions.

I would like to especially thank to my family for their excellent support and limitless patience.

I would like to express my appreciation to Irem YILMAZ for her understanding and support.

I would like to thank Kaan GÜRBÜZER for his excellent helping about wavelet functions, also I would like to thank Asst. Prof. Berkin MALKOÇ for the database he shared with me.

Mehmet Yavuz Onat

ANOMALY DETECTION IN TEMPORAL DATA MINING

ABSTRACT

Temporal data mining is interested in analyzing big amount of temporal data to automatically discover interesting patterns or relationships which gives useful information. Getting the valid, useful information is the main motivation of temporal data mining. The major tasks of temporal data mining are; indexing, clustering, classification, prediction, summarization, anomaly detection and segmentation. In temporal data, anomaly detection or novelty detection is the identification of interesting patterns. Several anomaly detection algorithms have been proposed in the literature. In this thesis, anomaly detection methods, HOT-SAX, WAT, PAV and MPAV are investigated and accuracy of these methods is tested with real and synthetic time series data sets. Also, symbolic family of temporal data representation techniques is examined and effectiveness of these representation techniques is tested in terms of anomaly detection. Time series data sets with tagged anomaly points were used for comparing anomaly detection and representation methods. R statistical programming language was used for application. In general HOT-SAX algorithm performed better than other anomaly detection algorithms and Trend-based symbolic representation technique had better accuracy than other representation techniques.

Keywords: Temporal data mining, anomaly detection, temporal data representation, TSX, SAX, HOT-SAX

ZAMANSAL VERİ MADENCİLİĞİNDE ANOMALİ TESPİTİ

ÖZ

Zamansal veri madenciliği büyük miktarda zamansal veri içerisinden otomatik olarak ilgi çeken desenleri veya ilişkileri bularak veri hakkında anlamlı ve geçerli bilgi vermeyi amaçlar. Endeksleme, kümeleme, sınıflama, tahmin etme, özetleme, anomali tespiti ve bölümleme zamansal veri madenciliğinde önde gelen uygulama alanlarıdır. Zamansal veri içerisinde anomali tespiti veya farklılık tespiti ilgi çekici desenlerin saptanması olarak tanımlanmaktadır. Literatürde birçok anomali tespiti algoritması bulunmaktadır. Bu tezde, HOT-SAX, WAT, PAV ve MPAV algoritmaları incelenmiş ve performansları gerçek ve sentetik zaman serileri kullanılarak test edilmiştir. Ayrıca, sembolik zamansal veri temsil yöntemleri incelenmiş ve bu yöntemlerin geçerliliği anomali tespiti açısından test edilmiştir. Genel olarak HOT-SAX algoritması diğer anomali tespiti algoritmalarına göre ve Trend-based Symbolic Approximation diğer veri temsil yöntemlerine göre daha başarılı sonuçlar vermiştir. Anomali tespiti algoritmalarının ve veri temsil yöntemlerinin performanslarının karşılaştırılmasında anomalileri bilinen zaman serileri kullanılmıştır. Uygulamada R istatistiksel programlama dili kullanılmıştır.

Anahtar kelimeler: Zamansal veri madenciliği, anomali tespiti, zamansal veri temsil yöntemleri, TSX, SAX, HOT-SAX

CONTENTS

Page

M.Sc. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES viii
LIST OF TABLES ix

CHAPTER ONE - INTRODUCTION 1

CHAPTER TWO – TEMPORAL DATA MINING 4

2.1 Temporal Data 4
2.2 Temporal Data Mining 5
2.3 Temporal Data Mining Tasks 6

CHAPTER THREE – TEMPORAL DATA REPRESENTATION TECHNIQUES 7

3.1 Piecewise Aggregate Approximation 8
3.2 Symbolic Aggregate Approximation 9
3.3 Extended SAX 11
3.4 Trend-based Symbolic Approximation 14
3.5 Adaptive Symbolic Aggregate Approximation 16
3.6 Haar Wavelet Transform 18

CHAPTER FOUR – ANOMALY DETECTION 22

4.1 Point Anomalies 24
4.2 Subsequence Anomalies 24

4.2.1 Heuristically Ordered Time Series using Symbolic Aggregate Approximation 25
4.2.2 Pattern Anomaly Value 26
4.2.3 Multi-Scale Abnormal Pattern Mining Algorithm 28
4.2.4 Wavelet and Augmented Trie 29

CHAPTER FIVE – APPLICATION 32

5.1 Comparison of Symbolic Representation Techniques 32
5.2 Anomaly Detection for Synthetic Time Series 35
5.3 Anomaly Detection for Financial Time Series 36
5.4 Anomaly Detection for Electrocardiography Data 38
5.5 Application of Yahoo Webscope Anomaly Detection Dataset 41

CHAPTER SIX – CONCLUSION 49

REFERENCES 51

APPENDICES 55

LIST OF FIGURES

Page

Figure 3.1 Time series representations 7
Figure 3.2 PAA representation of a time series 8
Figure 3.3 Solution for undividable problem of PAA..... 9
Figure 3.4 SAX representation..... 10
Figure 3.5 Standard normal distribution with breakpoints 11
Figure 3.6 SAX representation for US/YEN exchange rate 12
Figure 3.7 ESAX representation for US/YEN exchange rate 13
Figure 3.8 Three occasions of MP and MD in a segment..... 15
Figure 3.9 Main trend segments for different occasions..... 15
Figure 3.10 USD/TRY exchange rate 18
Figure 3.11 USD/TRY exchange rates with reduced dimensionality by Haar wavelet transform..... 19
Figure 4.2 Example of collective anomaly in ECG data..... 23
Figure 4.3 Subsequence anomaly in a time series..... 25
Figure 5.1 Synthetic time series with tagged anomaly points..... 33
Figure 5.2 Synthetic time series $X_1(t)$ and $X_2(t)$, $X_2(t)$ includes an anomaly sequence between 600 and 620. 35
Figure 5.3 EUR/USD exchange rates..... 37
Figure 5.4 EUR/USD exchange rates and red colored anomaly sequence 38
Figure 5.5 ECG data, anomaly sequence is colored with red 39
Figure 5.6 Red anomaly sequence found by HOT-SAX and WAT, HOT-SAX also found the blue anomaly sequence 40
Figure 5.7 ST waves extracted from the time series 41
Figure 5.8 Real time series plots..... 42
Figure 5.9 Synthetic time series plots 45

LIST OF TABLES

Page

Table 3.1 Angle breakpoint interval search table 16
Table 5.1 Parameter settings for HOT-SAX algorithm 34
Table 5.2 Results for representation techniques 34
Table 5.3 Results for synthetic time series 36
Table 5.4 Dates of anomaly sequences found by algorithms 37
Table 5.5 Results for ECG data 40
Table 5.6 Parameter settings for real time series application 43
Table 5.7 Results for real time series 43
Table 5.8 Parameter settings for synthetic time series application 46
Table 5.9 Results for synthetic time series 46
Table 5.10 General accuracy of algorithms 48

CHAPTER ONE
INTRODUCTION

Data mining is interested in analyzing big amount of data to automatically discover interesting patterns or relationships which gives useful information. Temporal data mining is concerned with such analysis executed over temporal data. In this thesis, common tasks of temporal data mining, particularly anomaly detection task, are investigated. Anomaly detection techniques are tested by performing algorithms using temporal data with different characteristics.

Temporal data mining gathers techniques from statistics, machine learning and database technologies. The goal is to discover unusual patterns, unexpected trends or subtle changes in temporal data (Laxman & Sastry, 2006). Temporal data refers to data, where changes over time or temporal aspects play a central role or are of interest. Temporal data can be seen in several forms; most common one is time series and most of the temporal data mining methods have been developed for time series data. However, temporal data doesn't have to be a collection of observations made chronologically. It can be a collection of sequences and any kind of temporal dependency between observations without chronological order can make it temporal data. Getting the useful, valid information from temporal data is the main idea of the temporal data mining.

A time series is a set of observations, usually collected at regular intervals. Time series data occur naturally in many application areas; economics, finance environmental, medical, etc. Analyzing time series has been an interest for researchers for a long time and it has been considered as whole different numeric field in statistics or computer science (Fu, 2011). Time series analysis has a long history, techniques for statistical modeling and spectral analysis of time series have been in use for more than fifty years (Box et al., 1994). Well-known approach for modeling is the Box-Jenkins method, named after the statisticians George Box and Gwilym Jenkins, applies autoregressive moving average Autoregressive Moving

Average (ARMA) or Autoregressive Integrated Moving Average (ARIMA) models to find the best fit of a time series model to past values of a time series.

In data mining, as a branch of computer science and statistics, temporal data mining is a title for data mining techniques executed over temporal data. New methods have been developing for temporal data mining since the late 1990s. Temporal data mining is appraised as more challenging field because of the structure of temporal data and chronological dependency between observations. Most of the works in temporal data mining are applied to the database of time series (Agrawal et al., 1993, Keogh et al., 2000). However, there are studies that use one time series for their work (Keogh et al., 1997). Temporal data mining tasks are the tools to get the information. Data representation, similarity measure and data mining methods are the essential components for these tasks. In this thesis, common temporal data representation techniques and temporal data mining task are examined.

Data representation is basically preparing the data for the data mining tasks. Dimensionality reduction and in some cases discretizing the data can be done by representation techniques. There are lots of benefits can be gained by working with discretized data. Calculation complexity can be decreased and the discretized data can form the hierarchical structure for hierarchical algorithms. Well-known methods, Fourier and wavelet transforms are widely used as a representation technique in many applications for temporal data. Also new approaches have been developed to represent temporal data.

Detecting anomalies has always been important for almost every field. In temporal data mining, anomaly detection task is an attractive field. Anomaly detection is an important problem that has been researched within diverse research areas and application domains (Chandola, 2004). In data mining, anomaly detection (or outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Data mining methods for detecting anomalies can change for different fields. In temporal data mining, the chronological dependence between observations can make algorithms and indexing

mechanisms can be more complicated. Nevertheless, anomalies are classified into two classes in temporal data mining for time series data; sequence anomalies and point anomalies. In this thesis; sequence anomaly detection techniques are examined.

Financial, medical and synthetic time series were used in application. Synthetic time series with different characteristics were preferred. Such characteristics are; seasonality, trend and noise. Representation methods and anomaly detection methods are compared with these time series.

Temporal data concept and most common tasks of temporal data mining are shortly briefed in Chapter-2. Representation techniques of temporal data are referred in Chapter-3, Anomaly detection tasks are discussed in detail in Chapter-4. Anomaly detection application and the discussion of the results are given in Chapter 5. Finally, in Chapter-6, some conclusions and suggestions for future works are offered. The R codes used in the study are given in the Appendix part.

CHAPTER TWO
TEMPORAL DATA MINING

Temporal data can be obtained easily from scientific and financial applications. A research by Tufte in 1983 shows that; a random sample of 4000 graphics from 15 of the world's newspapers published from 1974 to 1989 found that more than 75% of all graphics were time series (Tufte, 1983). Nowadays, temporal data takes up a large portion of world's data supply. Getting the useful, valid information from temporal data is the main idea of temporal data mining. In this chapter; temporal data types, goals of temporal data mining and temporal data mining tasks are given.

2.1 Temporal Data

Temporal data is data that varies over time and can be in several forms. Mitsa (2010) classified temporal data into 3 classes as following;

- Time series: They represent ordered real-valued measurements at regular temporal intervals. A time series $X = \{x_1, x_2, \dots, x_n\}$ for $t = \{t_1, t_2, \dots, t_n\}$ is a discrete function with value x_1 for time t_1, value x_2 for time t_2, and so on.

- Temporal sequences

These can be time-stamped at regular or irregular time intervals. An example of a temporal sequence is the time-stamped sequence of purchases of a customer on a Web site.

- Semantic temporal data

These are defined within the context of ontology. For example, "young" and "adult" are defined within the context of human lifetime ontology.

Since time series is the most common type of temporal data, most of the temporal data mining methods aim to analyze time series or time series database. But most of

the algorithms can be adapted to different types of temporal data by changing the indexing mechanism.

2.2 Temporal Data Mining

Temporal data mining is a collection of methods (statistics, machine learning, etc.) to get valid information from temporal data. Temporal data mining methods offer to extract variety of different information from the data, whereas classical time series analysis can offer limited information. Temporal data mining methods have been developing since the late 1990s, and its importance has been shown in many applications. Most of the works in temporal data mining analyze large volumes of medical time series (ECG, EKG), financial time series (stock prices, exchange rates, etc.) and many more different domains of temporal data.

Temporal data mining literature shelter applications from many different fields. For example, lake volumes in North America were investigated in Meyer's work (2005). Another interesting example can be the work of Jakkula and Cook in 2007; they used temporal pattern discovery methods to detect anomalies in a smart home. Review articles of the different authors, Ratanamahatana et al., (2005), Fu (2010), Esling et al., (2012), are the most important sources for the temporal data mining literature.

First main difference between the classic time series analysis and the temporal data mining lies in the size and nature of data sets. Temporal data mining methods often must be capable of analyzing data sets that are prohibitively large for conventional time series modeling techniques to handle effectively. Second main difference lies in the kind of information we want to get from the data. The scope of temporal data mining extends beyond the standard forecast or control applications (Laxman & Sastry, 2006).

2.3 Temporal Data Mining Tasks

There are various tasks of temporal data mining; most of them come with adapting known data mining tasks for temporal data. However, new approaches have been introduced. According to Ratanamahatana et al, 2010, major tasks of temporal data mining considered by the temporal data mining community are; indexing, clustering, classification, prediction, summarization, anomaly detection and segmentation.

- Indexing

In literature indexing can be found as query by content or similarity search. Indexing mechanisms of temporal data is considered more complicated than other data types because of the nature of temporal data. Indexing is sometimes considered a special case of data mining, and many data mining algorithms use indexing as a subroutine (Keogh, 2008)

- Clustering

Finding similar time series in a time series database or the natural groups in a given time series. For instance; if we want to find the similar stock prices for Dow Jones Industrial Average companies, we can apply clustering task under some similarity measure to find it.

- Classsification

General usage for classification task is to assign an unlabeled time series in a time series database to one of two or more predefined classes.

- Anomaly Detection

Anomaly detection refers to the problem of finding patterns in data that do not conform the expected behavior (Chandola, 2010). Variety of anomaly detection techniques have been developed to find these non-conforming patterns for temporal data. Lots of anomaly detection algorithms introduced with anomaly detection task for temporal data mining; HOT-SAX (Keogh, 2005), WAT (Fu, 2006), SVR-Based (Ma & Perkins, 2003) and many more.

CHAPTER THREE
TEMPORAL DATA REPRESENTATION TECHNIQUES

Time series are essentially high dimensional data. Applying any temporal data mining algorithm to the original raw data would be computationally too expensive. Dimension reduction is one of the major reasons for representing the original data. The main motivation of representations is thus to emphasize the essential characteristics of the data in concise way (Esling, 2012).

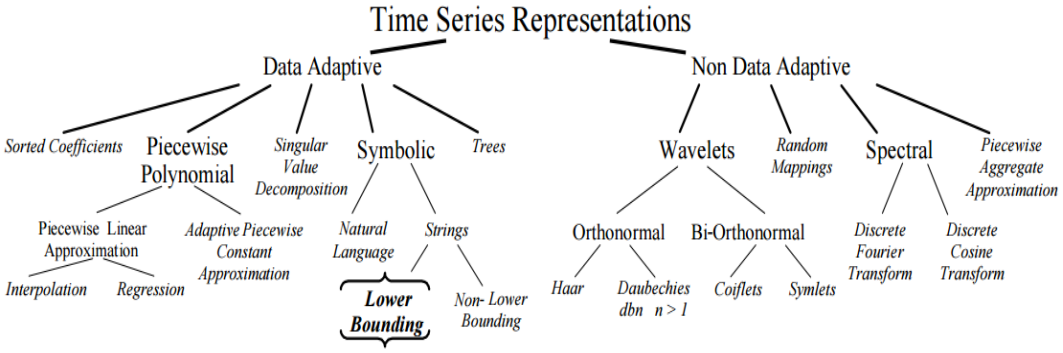


Figure 3.1 Time series representations (Keogh et al., 2005)

There are many data representation techniques for temporal data. Keogh et al. (2005) summarized some of the well-known data representation techniques in Figure 3.1. The first two branches of the figure are data adaptive and non-data adaptive. Non-data adaptive branch refers to the methods that parameters of the representation technique remain the same for every time series. Data adaptive method implies that the parameters of the transformation are modified depending on the data available.

Data representation does reduce the dimensionality but also provides information about the intended characteristics of time series. Trends, shapes and patterns contained within data are the common characteristics which are aimed to be captured with representation techniques. In this chapter, some of the temporal data representation techniques are investigated.

3.1 Piecewise Aggregate Approximation

Piecewise aggregate approximation (PAA) was independently suggested by Keogh (2001) and Yi & Faloutsos (2000). PAA reduces dimensionality by dividing time series in to the equal segments and recording the mean of each segment with following steps;

1. A time series C of length *n* can be represented in a *w*-dimensional space by a vector.
2. *w* segments are extracted from *C* time series and length of the segments is *n/w*.
3. the ith value of $\bar{C}$ where $\bar{C} = \bar{c}_1, \bar{c}_2, \dots, \bar{c}_w$ is calculated by the following equation;

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} C_j \qquad i = 1, \dots, w \tag{3.1}$$

Figure 3.2 illustrates the idea;

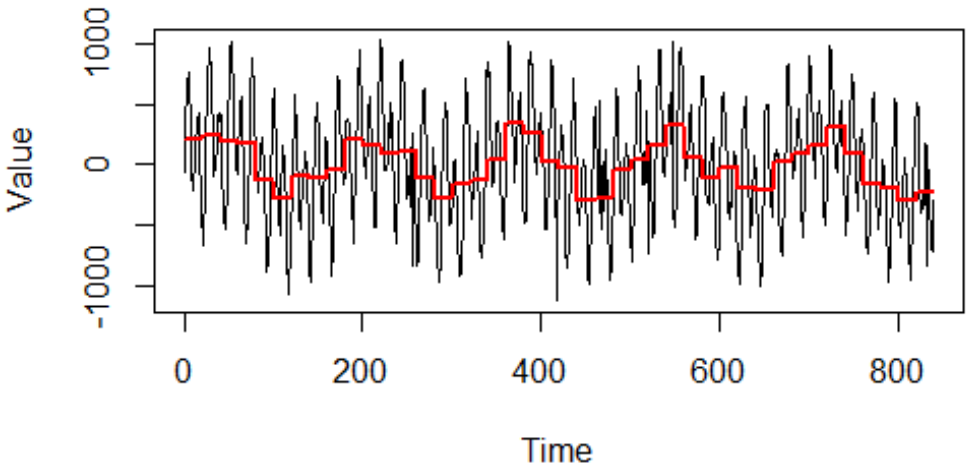


Figure 3.2 PAA representation of a time series

So far, PAA was described with the assumption that the length of the time series is divisible by the number of the segments. Lin et al (2003) provided a simple solution when n/w is not an integer, for instance in Figure 3.3-A: 10 data points are divided into 5 segments and n/w is an integer but in Figure 3.3-B: 10 data points are divided into 3 segments and point 4 and 7 contributes into two segments with weights shown in figure. This makes each segment contains exactly 3 1/3 data points and solves the undividable problem (Lin et al, 2003).

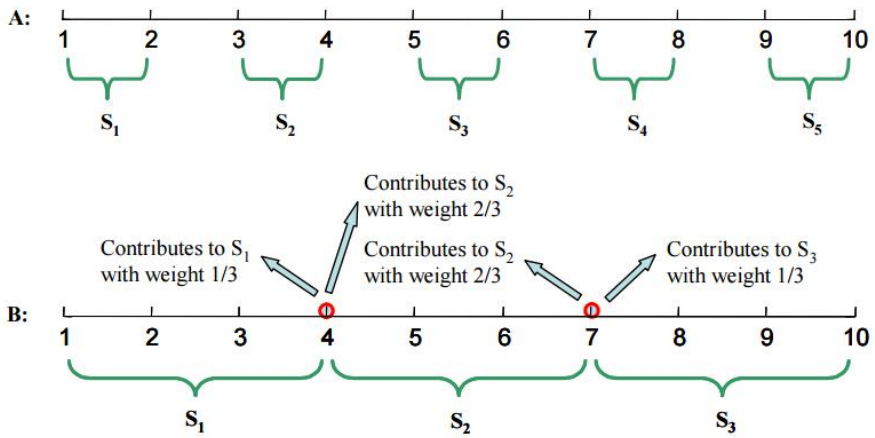


Figure 3.3 Solution for undividable problem of PAA

3.2 Symbolic Aggregate Approximation

Discretizing the time series with symbols is another family of time series representations. Symbolic Aggregate Approximation (SAX) is one of the most common symbolic representation techniques. SAX is introduced by Lin et al (2003), which has been shown to preserve meaningful information from the original data. Basic idea of SAX is to convert the time series into a discrete format, with a small alphabet size. SAX representation follows these steps;

1. Normalization: Transform the raw time series C into normalized C' with mean 0 and standard deviation of 1.

$$c_i^{'} = \frac{c_i - \bar{x}}{s} \qquad i = 1, \dots, n \tag{3.2}$$

2. Dimensionality reduction via PAA: Represent normalized time series C' with PAA and obtain $\bar{C}$ of length w.

3. Discretization: Use standard normal distribution to determine breakpoints and discrete the $\bar{C}$ into symbols depending on breakpoints.

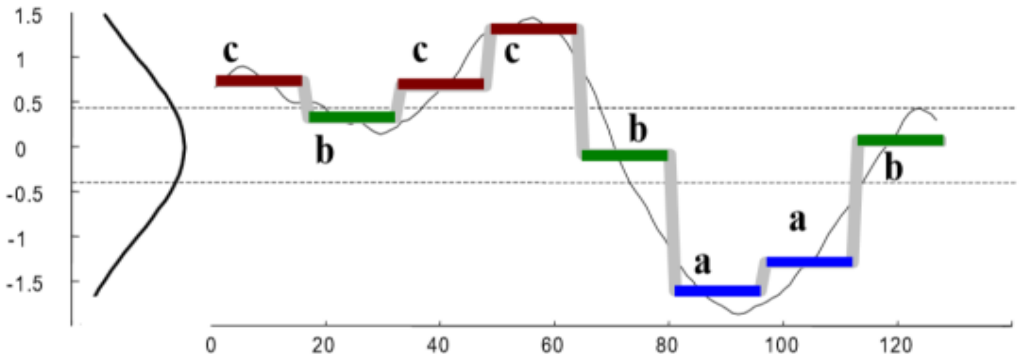


Figure 3.4 SAX representation (Lin et al, 2003)

SAX representation illustrated in Figure 3.4, colored lines represent the segments obtained from PAA. SAX only requires one parameter and that is alphabet cardinality. Alphabet cardinality stands for the letters of the alphabet which we want to represent the time series. In Figure 3.4; alphabet cardinality is three; “a”,”b”,”c” is determined by the researcher. Depending on the alphabet cardinality; standard normal distribution divided into equal intervals. To represent time series with alphabet cardinality *k*, standard normal distribution divided into *k* intervals, each interval has equal probability *1/k* and *k-1* breakpoints extracted from distribution. Figure 3.5 illustrates the idea.

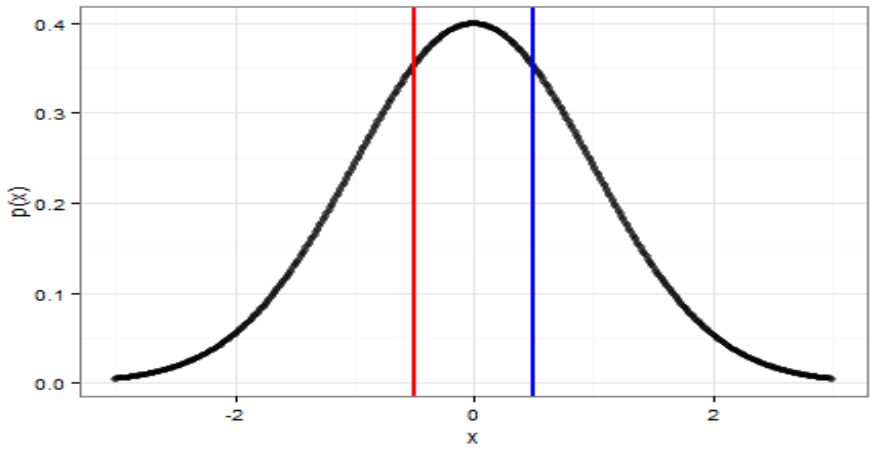


Figure 3.5 Standard Normal Distribution with breakpoints

In Figure 3.5; standard normal distribution divided into 3 equal intervals with equal probability 1/3. Three equal areas can be seen in Figure 3.5 with two breakpoints extracted. Breakpoints represent the 1/3 and 2/3 portion of the standard normal distribution. Let these breakpoints be β_1 and β_2, symbolic discretization for alphabet cardinality 3 is done by using the formula (3.3).

$$\text{Symbol}=\begin{cases} a; & \bar{c}_i < \beta_1 \\ b; & \beta_1 < \bar{c}_i < \beta_2 \\ c; & \bar{c}_i > \beta_2 \end{cases} \tag{3.3}$$

There are many symbolic representations of time series, most of them are based on SAX. Three of these representations (ESAX, aSAX, TSX) are briefed in this chapter.

3.3 Extended SAX

SAX is based on PAA representation for dimensionality reduction that minimizes the dimensionality by the mean values of each segment. This mean value based representation causes high possibility to miss some important patterns that might be important for financial time series. Extended Symbolic Aggregate approXimation (ESAX) is an approach based on SAX, it aims to realize efficient and accurate discovery of important patterns, necessary for financial applications. ESAX

introduced by Lkhagva, Suzuki and Kawagoe (2006) and ESAX can improve representation preciseness without losing symbolic nature of the original SAX representation.

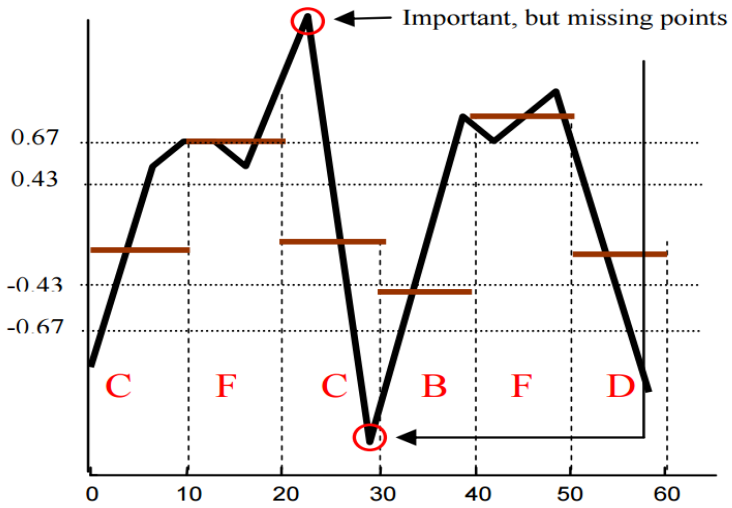


Figure 3.6 SAX representation for US/YEN exchange rate

For financial time series peak and dip points are important but as can be seen in Figure 3.6, SAX representation can miss that important patterns. The third segment is represented as “C” because; SAX uses the mean of each segment for dimensionality reduction. However, in the third segment there are two very important points which are shown in circle. In financial domains, those extreme points are very important. Extreme points in the third segment should be represented as “F”, the max point of the segment and “A”, the min point of the segment. Motivation for ESAX coming from that idea, ESAX approach uses two additional points, that is, maximum and minimum points, in equal sized frames besides the mean value for data approximation.

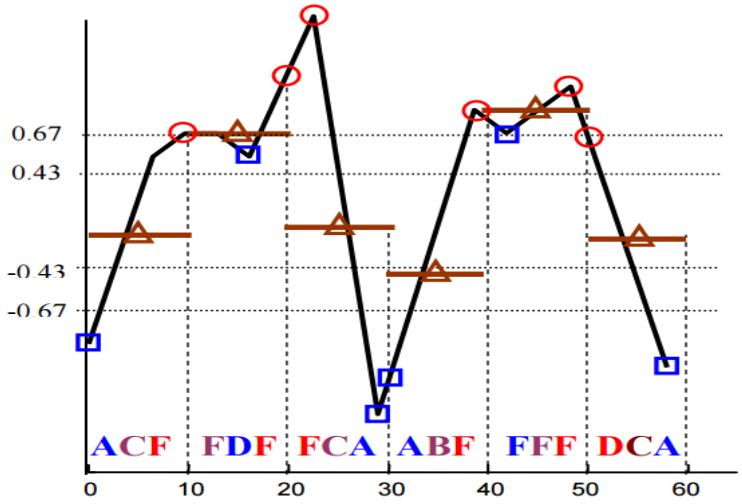


Figure 3.7 ESAX representation for US/YEN exchange rate

Same time series in Figure 3.6 is represented with ESAX in Figure 3.7. Time series is represented with additional max and min points besides mean value. Important thing to notice when discretizing with ESAX is the locations of the three values; let p_{min}, p_{mid} and p_{max} be the positions of max, min and mean values in the *k-th* segment on the time axis. Also S_{max}, S_{mid} and S_{min} are the symbolic representations of max, mean and min values of k-th segment.

$$p_{mid} = \frac{S_k + E_k}{2} \tag{3.4}$$

S_k and E_k are the starting and ending point on the x axis respectively. Ordering three symbols in *k-th* segment can be calculated by following expression (3.5).

$$< s_1, s_2, s_3 > = \begin{cases} < s_{max}, s_{mid}, s_{min} > if\ p_{max} < p_{mid} < p_{min} \\ < s_{min}, s_{mid}, s_{max} > if\ p_{min} < p_{mid} < p_{max} \\ < s_{min}, s_{max}, s_{mid} > if\ p_{min} < p_{max} < p_{mid} \\ < s_{max}, s_{min}, s_{mid} > if\ p_{max} < p_{min} < p_{mid} \\ < s_{mid}, s_{max}, s_{min} > if\ p_{mid} < p_{max} < p_{min} \\ < s_{mid}, s_{min}, s_{max} > \quad otherwise \end{cases} \tag{3.5}$$

The discretization of three values can be done as the similar way as the original SAX, which is defined in 3.3. A sequence of symbols is obtained after the discretization.

3.4 Trend-based Symbolic Approximation

SAX, ESAX and PAA representations suffer from the flaw or missing important trend feature, especially in financial area. Trend-based Symbolic Approximation (TSX) is a temporal data representation method developed by Li, Zhang and Yang in 2012. TSX reduces dimensionality and discretizes each segment into symbols considering its trend feature.

The main idea of TSX is to discretize mean and trend information into symbols respectively. Mean value of each segment is discretized by SAX and trend feature of segments are extracted by the slopes of three lines connecting four key points in each segment, and then discretized into symbols similar to SAX approach.

Key points of each segment are identified as following;

- Trend Line: The line connecting the start and the end of segment.
- The Most Peak Point (MP): The point which is above the trend line and has the largest distance to the trend line.
- The Most Dip Point (MD): The point which is below the trend line and has the largest distance to the trend line.

TSX evaluates vertical distance (VD) from the points to the trend line and determine whether points are MP or MD. The MP and MD in the segment are shown in Figure 3.8(a). By identifying MP and MD, there are four key points in the segment, the start, MP, MD and the end, respectively. According to the occurred order, connecting the four key points can extract three trend segments and thus obtain the slopes of three trend segments. Figure 3.9(a) illustrates the idea. However, not all occasions are identical. Not all segments have both MP and MD.

In Figure 3.8 and Figure 3.9 three different occasions are showed. If the segment only has MD as in Figure 3.8(b), key points of this segment are the start point, MD and the end point. First connect three points correctly and obtain the longer one and divide it into two equal segments, thus get three main trend segments. Figure 3.9(b) shows the idea of getting three trend segments when only MD occurs. Similar

process is applied when only MP occurs as in Figure 3.8(c) and getting the three main trend segments are shown in Figure 3.9(c).

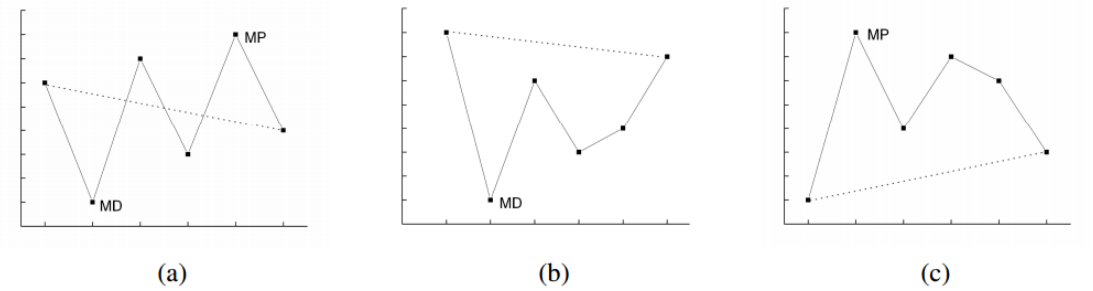


Figure 3.8 Three occasions of MP and MD in a segment

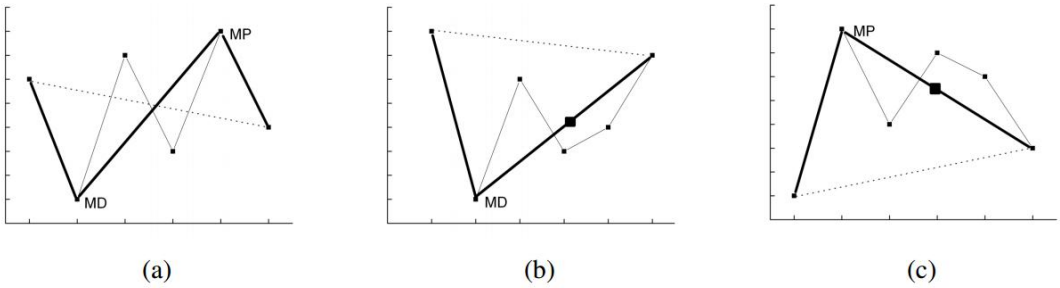


Figure 3.9 Main trend segments for different occasions

Discretization process of the trend segments is done by considering the slope value of each trend segment. The angle space corresponding to the slopes of trend segments is (-90°, 90°), which is divided into certain numbers of nonoverlapping intervals, each interval corresponds to a symbol similar to dividing standard normal distribution into equal intervals for SAX representation. Li et al., (2012) designed multiresolution angle breakpoint search table to symbolize the trend slopes.

Angle breakpoints (AB) are sorted list of angles, denoted by $AB = \theta_1, \ldots, \theta_{t-1}$ where t is the alphabet cardinality. Suppose each interval from $\theta_1$to θ_{i+1} has the equal probability of *1/t*, $\theta_0 = -\infty$, $\theta_0 = +\infty$.

Table 3.1 Angle breakpoint interval search table

θ_i	Trend alphabet cardinality (t)				
	2	3	4	5	6
θ_1	0°	-5°	-30°	-30°	-30°
θ_2		5°	0°	-5°	-5°
θ_3			30°	5°	0°
θ_4				30°	5°
θ_5					30°

TSX discretizes the slopes of three trend segments to symbols according to the angle breakpoint search table in Table 3.1. If the angle of trend segment is smaller than the smallest angle breakpoint θ_1, this trend segment is discretized into a symbol **"a"**; if the angle of trend segment is larger than the smallest angle breakpoint θ_1 and smaller than θ_2, this segment is mapped to symbol **"b"**; and so on.

Mean value of each segment is represented by SAX and trend segments are discretized into symbols according to Table 3.1. Thus, trend symbolic tuple of segment is obtained for each segment. Trend symbolic tuple of segment is denoted by a symbol tuple $\tilde{t}_i =< \tilde{t_i^1}, \tilde{t_i^2}, \tilde{t_i^3}, \tilde{t_i^4} >$ in this tuple $\tilde{t_i^1}$ is corresponding SAX representation of i^{th} segment. $\tilde{t_i^2}, \tilde{t_i^3}, \tilde{t_i^4}$ are corresponding to the three slopes in the i^{th} segment.

3.5 Adaptive Symbolic Aggregate Approximation

Adaptive Symbolic Aggregate Approximation (aSAX) is proposed by Pham, Le and Dang in 2010. aSAX is based on original SAX, but proposed adaptive vector of "breakpoints" different than SAX. These adaptive breakpoints determined by Llyod's algorithms which is nothing but *k*-means algorithm for clustering, specialized to one dimension.

aSAX offers an iterative method to determine the breakpoints. First breakpoints are determined by classical SAX approach; dividing standard normal distribution into equal sized frames with equal probability. After PAA representation, *n* size PAA vector is extracted. Let t_n denote the n^{th} PAA point in PAA vector. $\beta_0 = -\infty$ and $\beta_a = \infty$, $[\beta_i, \beta_{i+1})$ is an interval for $i = 0,\ldots,$a-1. For each interval new representative values are calculated. r_i denotes the representative value for the interval i. Set $\Delta = \infty$ and $\gamma > 0$. Algorithm can be summarized as follows;

1. For i = 0,…,a-1, compute the new representative value with the mean of all PAA points in the interval $[\beta_i, \beta_{i+1})$ by;

$$r_i = \frac{1}{N_i} \sum_{t_n \in [\beta_i, \beta_{i+1})} t_n \tag{3.6}$$

where, N_i is the total number of PAA points in the interval $[\beta_i, \beta_{i+1})$.

2. Compute new intervals by;

$$\beta_i = (r_{i-1} + r_i)/\,2 \tag{3.7}$$

for $i = 1,\ldots,$a-1

3. Compute the total representation error as;

$$\Delta' = \sum_{i=1}^{k} \sum_{t_n \in [\beta_i, \beta_{i+1})} (t_n - r_i)^2 \tag{3.8}$$

If $(\Delta - \Delta')/\Delta < \gamma$ then STOP. Otherwise, set $\Delta = \Delta'$ and go step 1.

The representation quality of each symbol directly affects the quality of symbolic representation with aSAX. Adaptive breakpoints produced better heuristic searching orders on not only the highly Gaussian distributed datasets but also the lack of Gaussian distribution datasets (Pham et al, 2010).

In the study of Pham et al, idea of having adaptive breakpoints for each segment is also discussed and they noted that quality of aSAX can be easily improved by doing that. This means that each adaptive breakpoints for each PAA segment, thus suffer more from processing and storage cost but great pruning power for discord detection.

3.6 Haar Wavelet Transform

Wavelets are basis functions used in representing data or other functions. The origin of wavelets can be traced to the work of Karl Weierstrass in 1873. Haar wavelet transform is widely used in different applications such as; signal - image processing and time series querying. The Haar sequence was proposed in 1909 by Alfréd Haar. Wavelet transformations aim to represent time series with wavelet functions and scale functions. The Haar wavelet is the simplest possible wavelet. Main benefit of Haar wavelet transform in temporal data mining is to reduce the dimensionality of data. In Figure 3.10 US Dollar versus Turkish Lira exchange rate is given for eight years, in Figure 3.11 same time series with reduced dimensionality using Haar wavelet transform is given. In Figure 3.10 time series contain 2550 observations and in Figure 3.1 second time series contain 318 data points and keeps the original characteristics of time series.



Figure 3.10 USD/TRY exchange rate

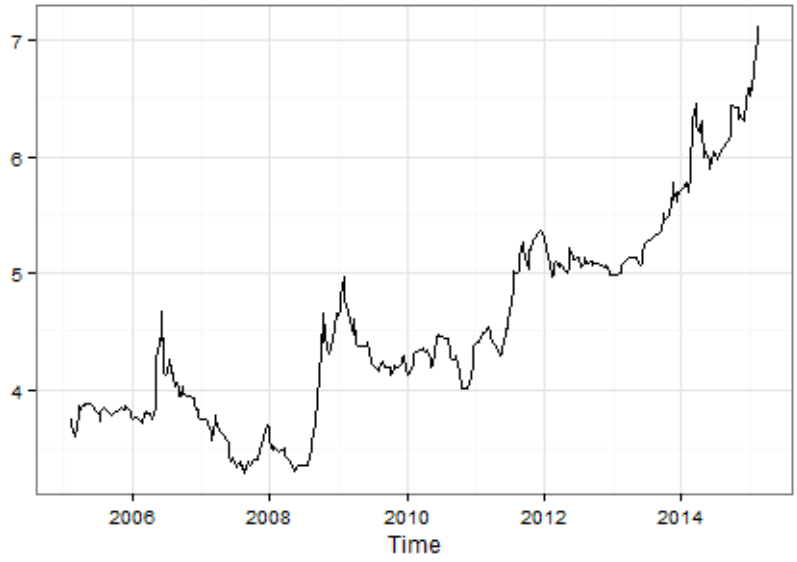


Figure 3.11 USD/TRY exchange rates with reduced dimensionality by Haar wavelet transform

Haar wavelet transform is a function that coefficients of function can represent time series, formulas are given in 3.10 and 3.11. Haar wavelet is a combination of wavelet function (3.9) and scale function (3.10). In literature; after dimensionality reduction with Haar wavelet transform, the time series can be normalized, so the scale problem can be solved (Bu et al., 2007).

Haar wavelet transform's mother wavelet function $\psi(t)$ is described as;

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & otherwise. \end{cases} \tag{3.9}$$

Scaling function $\phi(t)$ can be described as;

$$\phi(t) = \begin{cases} 1 & 0 \le t \le 1, \\ 0 & otherwise. \end{cases} \tag{3.10}$$

Haar transformation for a time series can be done by using Haar transform matrix. Scale and wavelet coefficients can be calculated and different resolutions can be

extracted with same matrix. Haar transform matrix changes for different sizes of *N*, for example for *N=2* Haar transform matrix is given in 3.11.

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{3.11}$$

For larger *N* size, new Haar transformation matrix is generated by placing the matrix given in 3.11 into diagonals (Wang, 2013). For example;

$$H_4 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix} \tag{3.12}$$

Scale coefficients are calculated for given time series $X = (x_1, x_2, \dots, x_n)$ as following;

$$\acute{x_\iota} = \frac{x_{2i+1} + x_{2i+2}}{\sqrt{2}} \quad (i = 0,1,2, \dots, \frac{n}{2} - 1) \tag{3.13}$$

The scaling function produces a smoother version of the time series, for first resolution it is half of the time series, further resolutions can be obtained by applying above process to wavelet coefficients. Wavelet coefficients calculated as;

$$d_i = \frac{x_{2i+1} - x_{2i+2}}{\sqrt{2}} \quad (i = 0,1,2, \dots, \frac{n}{2} - 1) \tag{3.14}$$

After calculating the scale and wavelet coefficients, matrix multiplication to describe Haar transform is given in 3.15 (Chen & Zhan, 2008).

$$W = \begin{bmatrix} x'_1 \\ d_1 \\ x'_2 \\ d_2 \\ . \\ . \\ x'_{n/2} \\ d_{n/2} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_{n-1} \\ x_n \end{bmatrix} \quad (3.15)$$

CHAPTER FOUR
ANOMALY DETECTION

There are many definitions of anomaly in the literature. One of the good definition of anomaly is made by Chandola (2014) and approved by many different field. Chandola defines anomalies as patterns in data that do not conform to a well-defined notion of normal behavior. The problem of finding these patterns is referred to as anomaly detection. Outlier, discord, exception, novelty, surprises, etc. are all referred as anomalies in different applications.

Anomalies are classified into three categories by Chandola;

- Point Anomalies

  Individual data instance can be considered as anomalous with respect to rest of data, and then the instance is termed as a point anomaly. Most of the point anomaly detection methods use modeling. Well-known approach is the Box-Jenkins ARIMA models for detecting point outliers. After fitting the best model, distances of each data point to fitted values with same time index calculated, Euclidian distance is mostly preferred for measuring the distance. Point outliers can be detected by considering each point's distance to fitted values.
- Contextual Anomalies

  If a data instance is anomalous in specific context, then it is termed as a contextual anomaly.
- Collective Anomalies

  Collection of data instances is anomalous with respect to the entire data; it is termed as a collective anomaly. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous. In temporal data mining, detecting sequence anomalies can be referred as detecting collective anomalies.

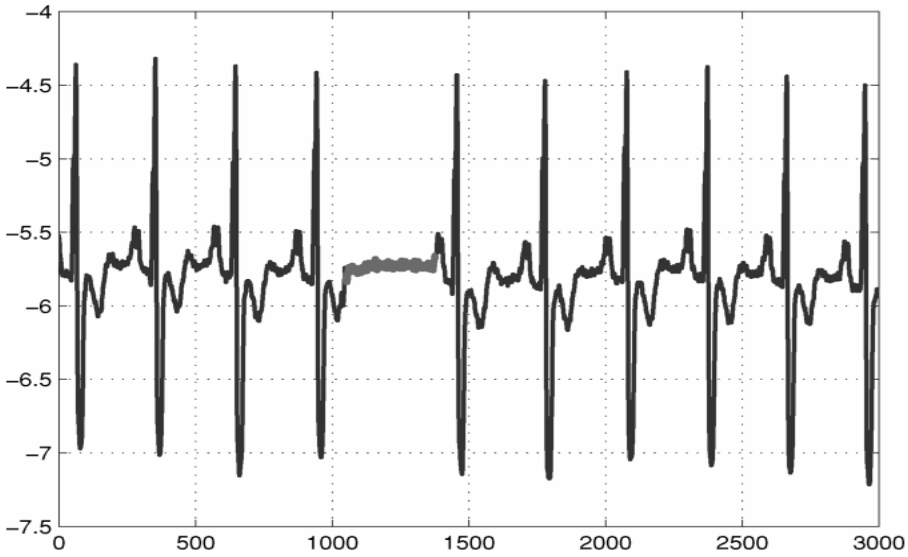


Figure 4.2 Example of collective anomaly in ECG data

Figure 4.2 illustrates the collective anomaly in electrocardiogram output, note that points in that sequence might not be anomalies by themselves. This sequence is an anomaly because it is different than the normal behavior.

In 2014, Gupta and et al. published a survey about outlier detection for temporal data. Their survey provides an extensive overview about anomaly detection methods for temporal data (Gupta et al, 2014). Gupta et al. investigated time series data with two perspectives; time series database and single time series. Temporal data mining tasks also can be applied to time series database by using the technique *whole series matching* and single time series by *subsequence matching* technique.

Anomaly detection methods for a single time series can be divided into two classes; point outliers and subsequence anomalies. In this thesis, detection of subsequence anomalies is focused. Detecting subsequence anomalies is a growing field; lots of algorithms have been introduced in this field. Most popular ones are; HOT-SAX, WAT, SVR-Based, PAV and MPAV. Further details of some of these approaches are briefed in this chapter.

4.1 Point Anomalies

Various methods have been proposed to find outlier points in a time series. Gupta et al., (2014) classified these approaches into three classes.

- Prediction Models

  The outlier score for a point in time series is computed as its deviation from predicted value by a summary prediction model.
- Profile Similarity based Approaches

  These approaches maintain a normal profile and then compare a new time point against this profile to decide whether it is an outlier.
- Deviants

  Deviants are outlier points in time series from a minimum description length point of view.

4.2 Subsequence Anomalies

Given a time series *T*, the subsequence *D* of length *n* beginning at position *l* is said to be the discord (anomaly) of *T* if D has the largest distance to its nearest non-self-match (Keogh et al, 2006). Another definition of subsequence anomalies is; combination of data points whose structure and frequency defies expectations. Subsequence anomalies are generally important in medical domains such as; an abnormal medical condition in a patient's heart can be detected by identifying anomalies in the time series corresponding to Electrocardiogram (ECG) recordings of the patient. Finding subsequence anomalies in financial domains also have been an interest for researchers, variety of representation techniques for financial time series have been introduced for anomaly detection task. Good representation of data is very important because most of the anomaly detection algorithms performance is highly depending on the representation of time series.

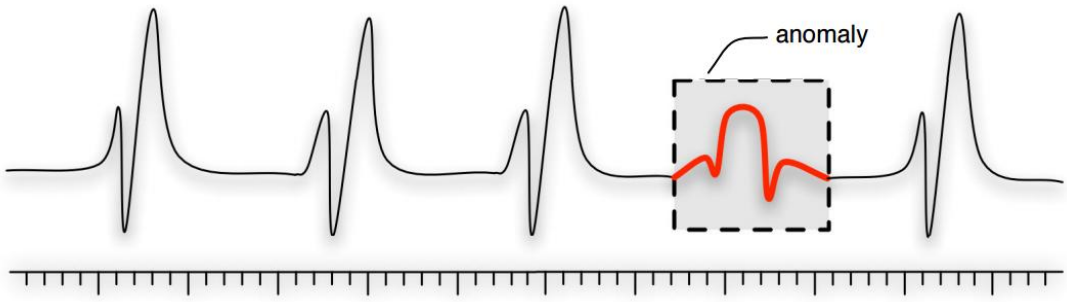


Figure 4.3 Subsequence anomaly in a time series (Esling, 2012).

4.2.1 Heuristically Ordered Time Series using Symbolic Aggregate Approximation

Heuristically Ordered Time series using Symbolic Aggregate Approximation (HOT-SAX) is a subsequence anomaly detection method introduced by Keogh, Lin and Fu in 2005. HOT-SAX is based on the similarity measures between subsequences and aims to find the most unusual subsequence in a time series. Only parameter that HOT-SAX needs is the length of subsequence (Keogh et al., 2005).

HOT-SAX extracts subsequences of time series by sliding window, which means that, all possible subsequences are extracted, for instance; let's consider a time series length *n* and subsequence length as *m*, we can extract *m-n+1* subsequences from the time series. Moreover, calculating the distances of these subsequences to all other non-self matches with heuristically adjustments is the main idea of HOT-SAX. Non-self match for HOT-SAX can be defined as; let *M* and *K* be the subsequences of time series length m and beginning positions of these subsequences are p and q respectively. If *M* is a non-self match of *K*, $|p - q| \geq m$ must be satisfied.

HOT-SAX algorithm follows these steps;

- Represent the time series by SAX
- Extract patterns by SAX words for each subsequence
- Based on the frequencies of patterns, determine anomaly candidates (Least observed pattern is more likely reveals the anomaly subsequence)
- Calculate the distance of anomaly candidates to all non-self matches.

- Highest distances to the nearest neighbors between anomaly candidates are marked as discords.

Finding distances for all subsequences with other subsequences might be more simple solution for finding the discords in time series but this idea is computationally more expensive than HOT-SAX. Obviously, this idea is not suitable for large datasets. Finding the anomaly candidates constructs heuristically structure for the algorithm and lowers the calculation cost.

HOT-SAX algorithm can be applied to any kind of symbolic representation of time series, other symbolic representations can represent time series in a same structure SAX does. Most of the symbolic representations of time series can be used for anomaly detection task with HOT-SAX approach. In this thesis, accuracy of symbolic representations for anomaly detection is tested with HOT-SAX approach.

4.2.2 Pattern Anomaly Value

Pattern Anomaly Value (PAV) is an anomaly detection method introduced by Chen and Zhan in 2008. PAV uses its own representation technique and represents the time series by linear patterns. Features of these linear patterns are slope and length of segments. Furthermore, depending on these representations, PAV calculates an anomaly value for each segment. Then, based on frequencies of anomaly values PAV algorithm detects anomalies (Chen & Zhan, 2008).

Chen and Zhan made four definitions for their algorithm;

Let X = $(x_1, x_2, \dots, x_n)$ be a time series where x_i is the amplitude at time t_i.

- **Linear pattern**; the linear segment joining two neighbor sampling points $<x_i, x_{i+1}>$ is named as linear pattern, simply pattern.
- **Support count**; Let $Y_i = <x_i, x_{i+1}>$ (i = 1,2,…,n-1) be the patterns of time series X. If the pattern Y_i occurs δ times in series X, the support count of pattern Y_i is defined as δ.

- **Anomaly value**; Let δ_i (i = 1,2,…, n-1) be the support count vector for each pattern and $\hat{\delta}_i$ be normalized to unit interval [0,1] using the formula;

$$\hat{\delta}_i = \frac{\delta_i - \min(\delta_i)}{\max(\delta_i) - \min(\delta_i)} \tag{4.1}$$

The anomaly value for Y_i is defined as;

$$AV_i = 1 - \hat{\delta}_i \tag{4.2}$$

- **Anomaly pattern**; pattern Y_i having $AV_i \geq minvalue$ is called anomaly pattern, where minvalue is the threshold of the anomaly value.

In some cases it is hard to determine an appropriate minvalue threshold, ranking the patterns on the basis of its AV decreasing order and declaring the greatest or top k patterns to find anomalies is suggested. Anomaly pattern can be defined as infrequent or rare patterns, in other words the lower support count increases the anomaly value of the pattern.

To determine whether the linear patterns are the same with other patterns; slope value and length of the pattern are considered. If the two patterns have same slope and length value, these two patterns are the same. By doing that, PAV algorithm represents the time series with linear patterns; each pattern (Y_i) has two features; slope (s_i) and length (l_i). Precision of the slope is the only parameter for PAV, this parameter is denoted by e and refers for the digits after decimal point when evaluating slope.

PAV algorithm summarized by Chen and Zhan as following;

- Compute the slope s_i and length l_i for pattern Y_i

$$s_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{4.3}$$

$$l_i = \sqrt{(x_{i+1} - x_i)^2 + (t_{i+1} - t_i)^2} \quad (4.4)$$

- Compare pattern Y_i with Y_j (i = 1,2,…,n-1; j = i,i+1,…,n-1), if the slope and length of Y_i and Y_j is equal, the support count of Y_i and Y_j is increased by 1. The vector $\delta = (\delta_1, \delta_2, \dots, \delta_{n-1})$ consists of the support counts of the patterns generated.

- Standardize the vector $\delta = (\delta_1, \delta_2, \dots, \delta_{n-1})$ by calculating the max and min points of support count vector. Obtain standardized values $\widehat{\delta} = (\hat{\delta}_1, \hat{\delta}_2, \dots, \hat{\delta}_{n-1})$.

- Compute the anomaly value AV_i of pattern Y_i, then generate and output the pattern anomaly vector $P = (AV_1, AV_2, .., AV_{n-1})$.

- Sort the anomaly value vector *P* in descending order and output the top *k* patterns or the patterns whose anomaly values are greater or equal to minvalue threshold.

4.2.3 Multi-Scale Abnormal Pattern Mining Algorithm

Multi-scale abnormal pattern mining algorithm (MPAV) is based on PAV and introduced by Chen and Zhan in 2008, they proposed PAV and MPAV in the same paper. They adopt the multiple levels Haar transform to reduce the dimensionality of time series and detect the anomaly pattern using PAV.

MPAV algorithm is summarized by the authors as following;

- The set of approximation coefficients (Haar wavelet transformation's scale function) A_k is obtained by *k* levels wavelet decomposition of time series X.
- The anomaly value sequence $(AV_1^k, AV_2^k, .., AV_{(\frac{n}{2^k}-1)}^k)$ of all patterns of new series $A_k$was calculated using the PAV algorithm.

- Reconstruct the sequence of anomaly values of A_k, that is calculate the pattern anomaly value of the original time series using formula;

$$AV_i = AV^k_{(\left[\frac{i-1}{k}\right]+1)} \quad i = 1,2,\dots,n-1 \tag{4.5}$$

- According to the anomaly value in descending order; top *k* patterns are the anomaly patterns, or the pattern whose abnormal value is larger than given threshold *minvalue.*

The first advantage of MPAV is that the wavelet transform can eliminate noise and reduce dimensionality of time series. Therefore, the accuracy and efficiency of the anomaly detection can be improved. Second advantage is that the multi-scale anomaly patterns can be detected from the compressed time series with different compressed ratio.

4.2.4 Wavelet and Augmented Trie

Wavelet and Augmented Trie (WAT) is based on Haar wavelet transform and aims to mine discords from time series. WAT was introduced by Fu, Leung, Keogh and Lin in 2006. WAT approach benefits from Haar wavelet transform's resolution flexibility. By using that flexibility, WAT algorithm does not require a word size parameter. WAT only requires two parameters; length of discord and alphabet size for discretization (Fu et al., 2006).

WAT approach basically follows these steps;

- Extract subsequences with sliding window
- Transform sequences with Haar wavelet transform starting with lowest resolution
- Discrete sequences in to a sequence (word) of finite symbols (construct patterns)
- Repeat this until a word appears only once otherwise increase the resolution of Haar wavelet transform

By following the process above, WAT dynamically adjust the effective word length according to the data characteristics. Transforming Haar wavelet coefficients into symbols is done with the similar idea of SAX. It is assumed that; for all i, the ith coefficient of all Haar wavelets in the same database tends to be evenly distributed between its minimum and maximum value, so cutpoints are determined by partitioning this region into several equal segments.

For the i^{th} coefficient, cutpoints are a sorted list of numbers $\beta_i = \beta_{1,i},\ \beta_{2,i}, \dots, \beta_{m,i}$ where *m* is the number of symbols in the alphabet and

$$\beta_{i,j} - \beta_{i,j+1} = \frac{\beta_{i,a} - \beta_{i,0}}{a} \tag{4.6}$$

$\beta_{i,0}$ and $\beta_{i,a}$ are defined as the smallest and the largest possible values of the ith coefficient, respectively. If the ith coefficient from Haar wavelet is in between $\beta_{i,1}$ and $\beta_{i,2}$, it is mapped to the first symbol "a". If the ith coefficient is between $\beta_{i,j-1}$ and $\beta_{i,j}$, it will be mapped to the symbol jth symbol, etc. In this way word for each subsequence is formed.

Length of discord is one of the parameters of WAT, but the word size is changing and dynamically adjusted by using different resolutions of Haar wavelet transform. For example, if discord length is determined as *m* the *n-m+1* subsequences extracted from a time series length *n*. (*n/2)* number of coefficients can be extracted with highest resolution of Haar wavelet transform This leads each subsequence represented by m/2 points so, m/2 word length. WAT starts from lowest possible resolution and increases the resolution till a word appears only once.

WAT was first proposed with the idea of finding the most unusual time series discord. However, algorithm was improved and aimed to find top-K discords in a given time series. Only the breakpoints approach has changed and instead of calculating a breakpoint for each Haar wavelet coefficient, they assumed the

coefficients are in Gaussian distribution and symbolic representation is done by SAX. The distribution of coefficients may affect the performance of algorithm but not the correctness. (Bu et al., 2007) WAT algorithm makes use of some valuable properties of Haar wavelet transform and is close to the ideal parameter free data mining, although, finding top-K discords has a lot of advantages then finding the most unusual discord.

CHAPTER FIVE
APPLICATION

In this chapter, accuracy of anomaly detection algorithms and representation techniques are tested with some real and synthetic data sets. Real data sets contain financial and medical time series. Also, Yahoo Webscope Anomaly Detection Dataset was used, datasets downloaded from http://webscope.sandbox.yahoo.com/ (Yahoo Webscope, 2015). Symbolic temporal data representation techniques are applied to time series which have different characteristic (seasonality, trend and both). Effectiveness of the representation techniques with HOT-SAX algorithm is examined. Euclidian distance is preferred for measuring similarity for all algorithms. R statistical programming language was used for application. R codes used in the study were given in Appendix.

5.1 Comparison of Symbolic Representation Techniques

In this section, symbolic representation techniques in Chapter-3 (SAX, ESAX, aSAX, TSX) are compared in terms of anomaly detection. Three synthetic time series with different characteristics are used in application. Seasonality, trend and both of them are considered as such characteristics. Time series have tagged anomaly points. HOT-SAX algorithm is modified for this application by adapting different symbolic representation techniques. TSX and ESAX require one sequence to construct patterns. However, SAX and aSAX require more than one sequence; in other words, both representation techniques are built on PAA representation. Parameter settings are decided considering these variations. Subsequence length, word length and alphabet cardinality are given in Table 5.1. Time series plots used in the application are given in Figure 5.1. All time series include noise and first time series consist one seasonality feature, second time series consist trend and third time series consist two different seasonality and trend together. Results for anomaly sequences in time series are given in Table 5.2. Coverage accuracy (5.1) used to calculate accuracy of representation techniques. Coverage accuracies are given in the second rows of Table 5.2.

$$Coverage\ Accuracy(CA) = \frac{Common\ points\ with\ anomaly\ sequence}{Length\ of\ anomaly\ sequence} \quad (5.1)$$

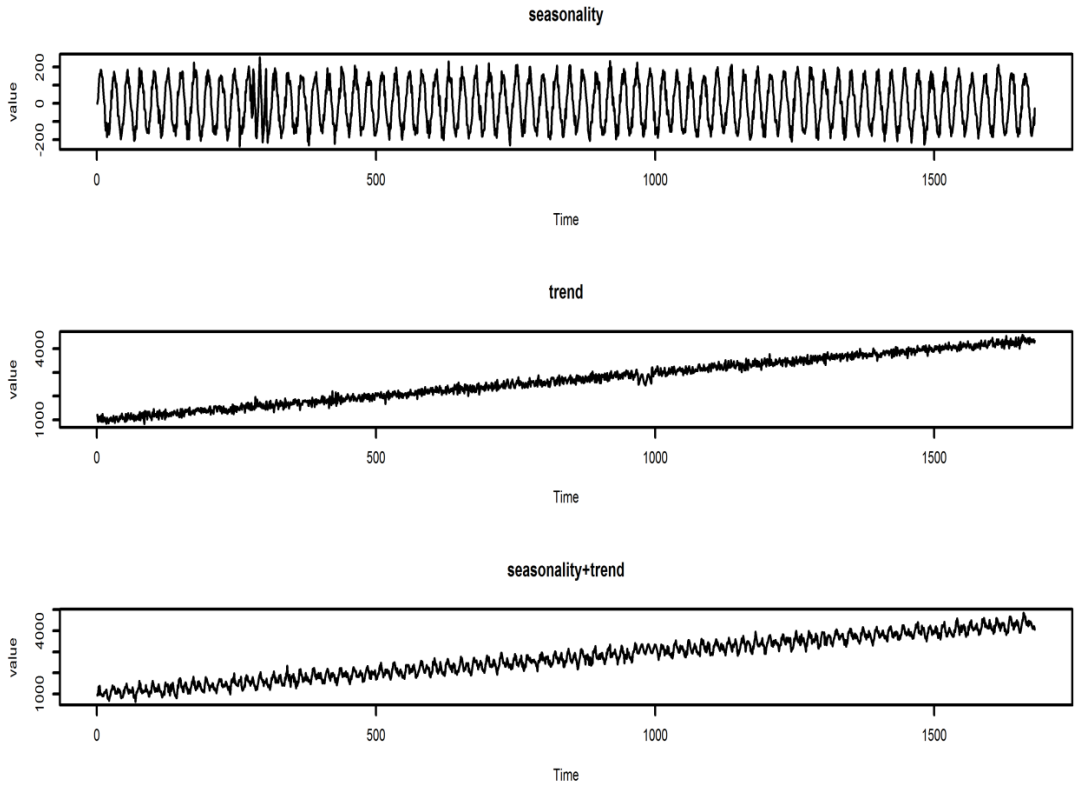


Figure 5.1 Synthetic time series with tagged anomaly points

Anomaly sequences found by adapted HOT-SAX algorithm are given in Table 5.2. Under seasonality feature, SAX and aSAX representation techniques had 62.5% coverage accuracy, whereas TSX had 100% coverage accuracy. However, ESAX didn't manage to find any anomaly sequences. Under trend feature, TSX gave significantly better results with 95.8% coverage accuracy. Third time series consist both trend and seasonality feature and again TSX gave significantly better results than other representation techniques. ESAX technique failed to represent time series in terms of finding anomalies, however, this representation technique designed for especially financial time series. Financial time series might have different characteristics considered in this application. In general, TSX technique gave better results than other representation techniques.

Table 5.1 Parameter settings for HOT-SAX algorithm

		Seasonality	Trend	Seasonality + trend
SAX	Subsequence Length	24	24	36
	Word Length	4	4	4
	Alphabet Cardinality	5	5	5
ESAX	Subsequence Length	24	24	36
	Alphabet Cardinality	5	5	5
TSX	Subsequence Length	24	24	36
	Alphabet Cardinality	4	4	4
aSAX	Subsequence Length	24	24	36
	Word Length	4	4	4
	Alphabet Cardinality	5	5	5

Table 5.2 Results for representation techniques

	Time series-1 Seasonality	Time series-2 Trend	Time series-3 Seasonality + Trend
Anomaly Sequence Index	280:303	970:993	970:1005
SAX	289:312 (62.5%)	951:974 (20.8%)	999:1034 (19.4%)
ESAX	- (0%)	- (0%)	- (0%)
TSX	280:303 (100%)	969:992 (95.8%)	971:1006 (97.2%)
aSAX	289:312 (62.5%)	988:1011 (25%)	- (0%)

5.2 Anomaly Detection for Synthetic Time Series

Synthetic time series includes two time series generated from the following stochastic process;

$$X_1(t) = \sin\left(\frac{40\pi}{N}\ t\right) + n(t) \tag{5.1}$$

$$X_2(t) = \sin\left(\frac{40\pi}{N}\ t\right) + n(t) + e_1(t) \tag{5.2}$$

t=1,2,3,…,N and N=1200. $n(t)$ is an Additive Gaussian White Noise with mean and standard deviation are 0 and 0.1 respectively. $e_1(t)$ is an anomaly sequence defined in 5.3 (Ma & Perkins, 2003).

$$e_1(t) = \begin{Bmatrix} n_1(t), & t \in [600,620] \\ 0 & otherwise \end{Bmatrix} \tag{5.3}$$

Where $n_1(t)$ is a normal distribution with mean 0 and standard deviation 0.5.

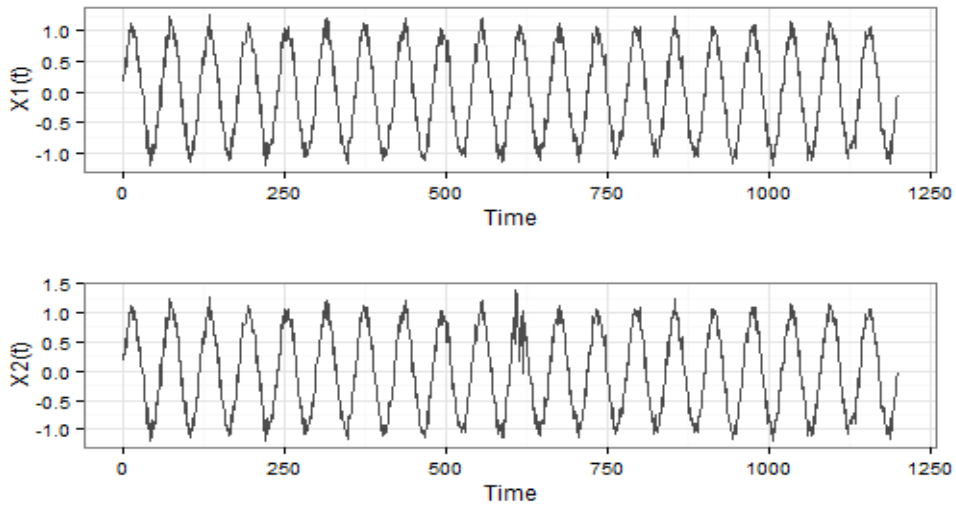


Figure 5.2 Synthetic time series $X_1(t)$ and $X_2(t)$, $X_2(t)$ includes an anomaly sequence between 600 and 620.

Parameters for this application are set as following; for WAT and HOT-SAX subsequence length and alphabet cardinality is set to 20 and 4, respectively. Word length for HOT-SAX is set to 5. For MPAV and PAV parameter is set to 2; 2 decimal digits. Also for MPAV, Haar wavelet resolution is set to 2. Anomaly sequences found by algorithms are given in Table 5.3.

Table 5.3 Results for synthetic time series

	Anomaly sequence index
Anomaly	600:621
HOT-SAX	601:620
WAT	601:620
PAV	600:620
MPAV	600:634

All anomaly detection algorithms found the anomaly sequence between 600 and 620. However, PAV and MPAV reveal other anomalous events but not as a valid sequence, across all anomalous events only the sequences given in Table 5.3 can be considered as a sequence.

5.3 Anomaly Detection for Financial Time Series

Anomaly detection algorithms; HOT-SAX, WAT, PAV and MPAV are applied to Euro – Dollar exchange rates between December 2003 – December 2014. Data is gathered from European Central Bank web site (https://www.ecb.europa.eu). Time series contains 2820 exchange rates recorded on work days. This is a real data and the anomaly patterns are not known in advance. There are no tagged anomaly points. Time series is given in Figure 5.3.

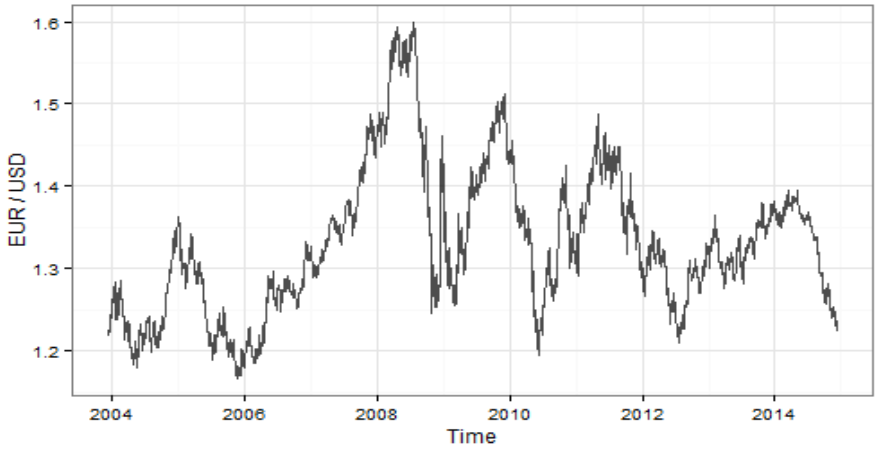


Figure 5.3 EUR/USD exchange rates

Parameters of algorithms are determined for the time series. In this case we are seeking for an anomaly in a 30 days period. For HOT-SAX and WAT; subsequence length is set to 30 and for symbolic representation, alphabet cardinality is set to 4 (a,b,c,d). Moreover, word length is set to 3 for HOT-SAX; which means PAA segment length is determined as 10 and each 30 day periods are represented with three letters with SAX. For MPAV and PAV parameter is set to 2; 2 decimal digits. Also for MPAV, Haar wavelet resolution is set to 2. Results are given in Table 5.4

Table 5.4 Dates of anomaly sequences found by algorithms

	Start	End
HOT-SAX	13-02-2009	26-03-2009
	16-02-2009	27-03-2009
	04-10-2010	12-11-2010
WAT	10-11-2008	19-12-2008
	24-02-2009	06-04-2009
PAV	16-12-2008	18-12-2008
MPAV	29-09-2008	21-10-2008
	10-12-2008	16-12-2008

HOT-SAX algorithm found three anomaly sequence and two of the sequences are 96.6% the same. This sequence had a huge chance to be an anomaly. WAT found

two anomaly sequences and one of them (24-02-2009 : 06-04-2009) occurs 5 days later than the anomaly sequence found by HOT-SAX, so %80 of the anomaly sequence is the same. Anomaly sequence is shown in Figure 5.4. Reason of this anomaly is traced back and it is found that FED made an unexpected declaration on second week of February, 2009. They declared that they will buy 300 billion dollars bond from the government. (Türkiye Bankalar Birliği, Bankacılık ve Araştırma Grubu, 2009). This might be the reason for falloff of the dollar price during the time sequence where anomaly found. PAV and MPAV algorithms found out different anomaly sequences than HOT-SAX and WAT. Nevertheless, all other anomaly sequences found by all algorithms might be anomalies and needs investigated.



Figure 5.4 EUR/USD exchange rates and red colored anomaly sequence

5.4 Anomaly Detection for Electrocardiography data

Electrocardiography (ECG) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. Anomaly detection methods are compared for the same ECG data, downloaded from UCR Time Series Database (Keogh et al, 2011). In the study of Keogh et al, it is noted that a cardiologist detected an anomaly in this ECG data. The ECG data is shown in Figure 5.5 and anomaly marked in red (Keogh et al, 2005). Anomaly sequence is a sign for premature ventricular contraction and area where anomaly found is called ST wave.

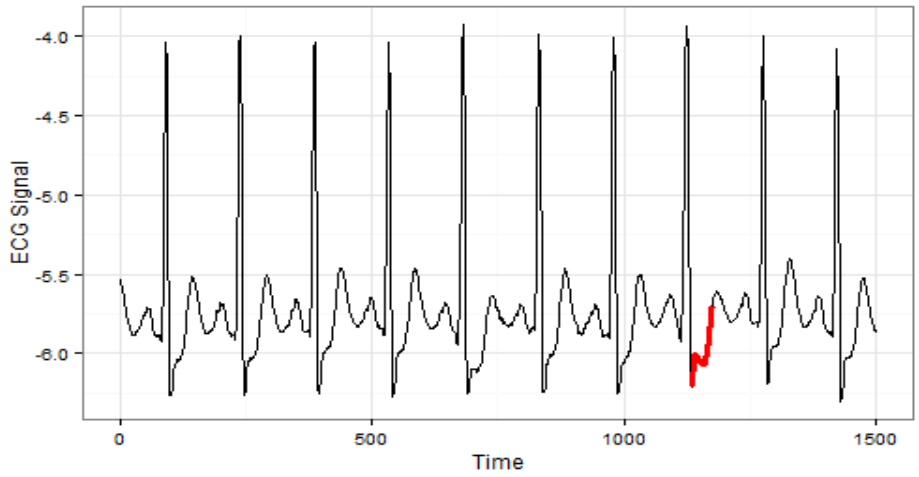


Figure 5.5 ECG data, anomaly sequence is colored with red

For this time series, parameters of HOT-SAX are determined as follows; sequence length=40, alphabet cardinality=5 and word length=4. HOT-SAX algorithm found four anomaly sequences and two of them cover 85% and 87.5% of the anomaly sequence marked by cardiologist. However, other two anomaly candidates pointed out sequences [682:721] and [683:722] (Blue marked in Figure 5.6). Sequence between 682 and 722 was investigated and it was found that 87.5% of the sequence belongs to the ST wave. In Figure 5.7, ST waves of these two anomaly sequences are compared with normal ST waves extracted from time series. Consequently, sequence between 682 and 722 is not an anomaly sequence, reasons are discussed later in this chapter.

WAT algorithm found 10 anomaly sequences and two of them cover 90% of the anomaly sequence revealed by cardiologist. Three of them cover 12.5% of the true anomaly sequence. Other 5 subsequences don't have common points with the anomaly sequence. PAV and MPAV algorithms couldn't find the true anomaly sequence. Results for all algorithms are given in Table 5.5.

Table 5.5 Results for ECG data

	Anomaly Sequence Index
Anomaly	**1132:1171**
HOT-SAX	1126:1165 (85.0%)
	1127:1166 (87.5%)
	682:721*
	683:722*
WAT	1128:1167 (90.0%)
	1136:1175 (90.0%)
PAV, MPAV	-

*Sequences further investigated, didn't considered as anomaly sequences.

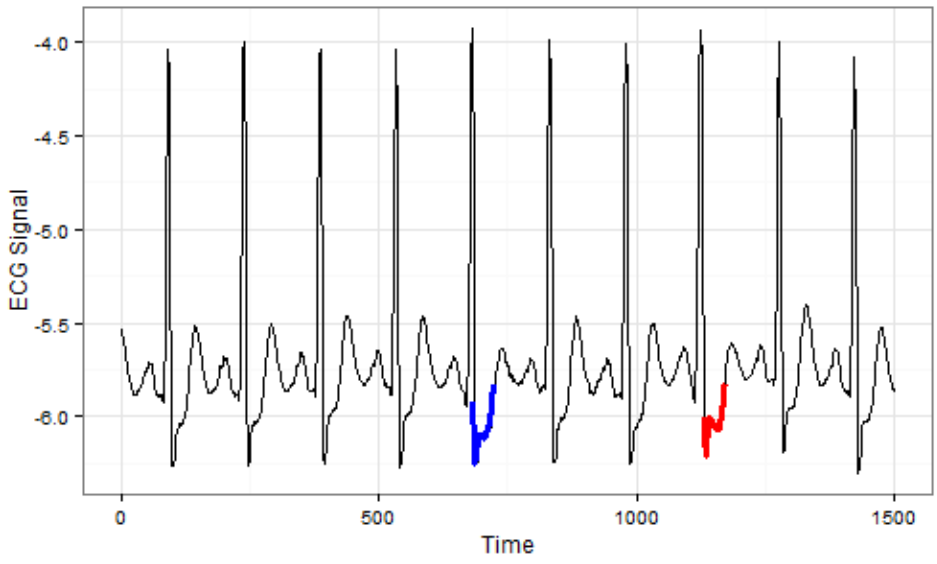


Figure 5.6 Red anomaly sequence found by HOT-SAX and WAT, HOT-SAX also found the blue anomaly sequence

ST waves are given in Figure 5.7. First sequence is the one cardiologist marked and also found by HOT-SAX and WAT. Second sequence is the ST wave of second anomaly found by HOT-SAX. Other four ST waves are extracted from the time

series. In the four normal ST waves, after the brief descending section, the signal rises monotonically. However, the anomalous ST wave (1st sequence) has an additional local peak caused by a premature beat, thus justifying the cardiologist's diagnosis of premature ventricular contraction (Keogh et al, 2005). 2nd sequence shouldn't be considered as an anomaly because it has the same characteristics with other 4 normal sequences. In general, for ECG data HOT-SAX and WAT gave significantly better results than MPAV and PAV. But, both algorithms couldn't find the discord directly like synthetic time series application.

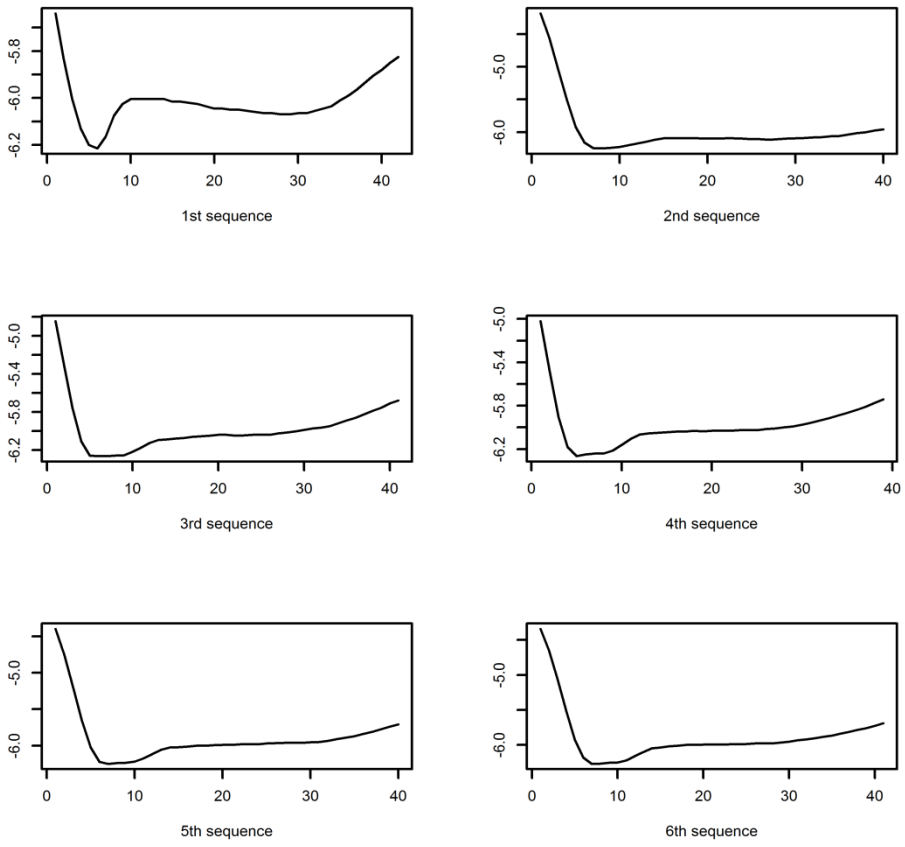


Figure 5.7 ST waves extracted from the time series

5.5 Application of Yahoo Webscope Anomaly Detection Dataset

For this application; 8 real time series and 10 synthetic time series were taken from Yahoo Webscope A Labeled Anomaly Detection Dataset, version 1.0. (http://webscope.sandbox.yahoo.com/). The goal of this dataset is to benchmark anomaly detection algorithms. The dataset consists of real and synthetic time-series

with tagged anomaly points. The synthetic data set consist of time series with varying trend, noise and seasonality. The real data set consist of time series representing the metrics of various Yahoo services. We tried to benchmark the anomaly detection methods using different characteristics of time series.

Coverage accuracy is calculated for each anomaly candidate, and then general accuracy of anomaly detection algorithm is calculated with mean of best coverage accuracies for time series.

In this chapter, HOT-SAX, WAT, PAV and MPAV algorithms are compared in terms of general accuracy.

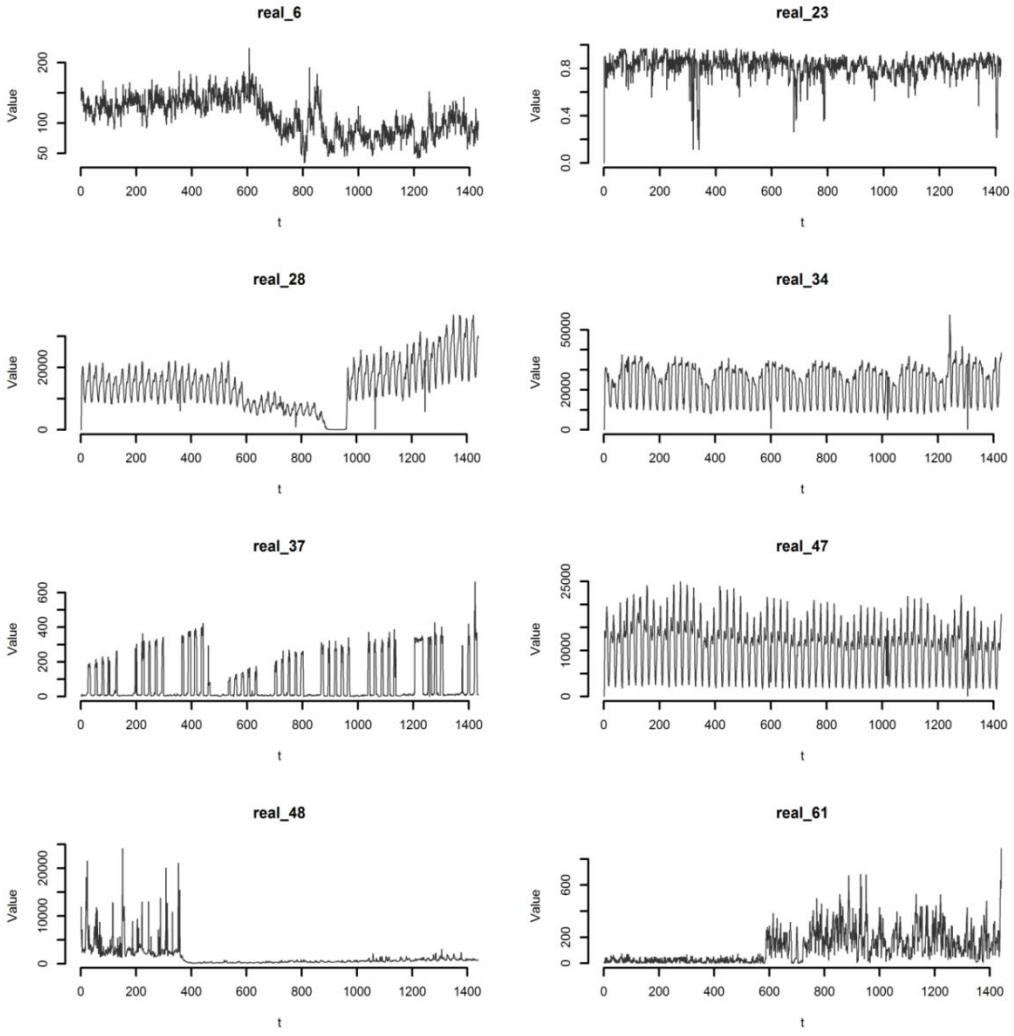


Figure 5.8 Real time series plots

Table 5.6 Parameter settings for real time series application

	Parameters	real_6	real_23	real_28	real_34	real_37	real_47	real_48	real_61
HOT-SAX	Subsequence length	15	24	80	9	30	8	24	24
	Word length	4	4	4	3	5	4	4	4
	Alphabet cardinality	3	5	5	5	5	5	5	5
WAT	Word length	15	24	79	9	30	7	24	24
	Alphabet cardinality	4	6	5	5	5	5	6	6
PAV	Precision of the slope	1	1	1	2	2	1	1	1
MPAV	Haar Wavelet Resoultion	2	2	2	3	2	2	2	2
	Precision of the slope	1	1	1	1	1	1	1	1

Table 5.7 Results for real time series

	real_6	real_23	real_28	real_34	real_37	real_47	real_48	real_61
Anomaly Index	821:835	315:341	885:963	1242:1250	1205:1235	592:600 1014:1020	363:383	588:611
HOT-SAX	822:836 **(93.3%)**	314:337 (88%)	923:1002 (51.5%)	1243:1251 **(88.8%)**	1201:1230 **(86.6%)**	594:601 **(87.5%)** 1017:1024 (57.1%)	348:381 (91.1%)	577:600 **(83.3%)**
	817:831 (73%)	317:340 (88.8%)	-	1244:1252 (77.7%)	-	595:602 (66.6%) 1018:1025 (42.8%)	347:370 (45.8%)	-
WAT	813:827 (46.6%)	297:321 (24.0%)	911:898 **(67.0%)**	1243:1251 (77.7%)	1211:1240 (83.3%)	595:601 (66.6%) 1014:1020 **(100%)**	351:374 (62.5%)	577:600 (83.3%)
	818:832 (80%)	-	927:1005 (46.8%)	-	-	-	359:382 **(95.8%)**	582:605 (75%)
PAV	822:826 (33.3%)	314:341 **(89.2%)**	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	600:620 (50%)
MPAV	816:844 (51.7%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	584:632 (24.4%)
	804:836 (45.5%)	-	-	-			-	604:624 (38.8%)

Eight real time series used in application are given in Figure 5.8 and parameter settings for each time series are given in Table 5.6. Results are given in Table 5.7. In general; HOT-SAX and WAT algorithms managed to find anomalies for all real time series, whereas PAV and MPAV had 3 and 2 successful findings respectively. First time series “real_6” includes one anomaly sequence and HOT-SAX had the best result with 93.3% coverage accuracy. PAV algorithm had the best result for

“real_23” time series with 89.2% coverage of anomaly sequence. “real_28” has an anomaly sequence of length 79, and WAT had the best result with coverage accuracy 67.0%. For time series “real_34”, HOT-SAX algorithm had the best coverage accuracy and revealed %88.8 of the anomaly sequence. “real_37” time series had an obvious anomaly sequence between 1205:1235, here HOT-SAX gave the best result again with coverage accuracy 86.6%. “real_47” time series includes two anomaly sequences, for the consistency for parameter setup, length of smaller anomaly sequence preferred for subsequence length parameter for HOT-SAX and WAT, as a result; WAT algorithm found the second anomaly sequence with 100% coverage accuracy, and best result for the first anomaly sequence found by HOT-SAX. Anomaly sequence in time series “real_48” found by HOT-SAX and WAT but WAT had the best coverage accuracy with 95.8%. Anomaly sequence in time series “real_61” found by all algorithms, however HOT-SAX had the best accuracy with 83.3%.

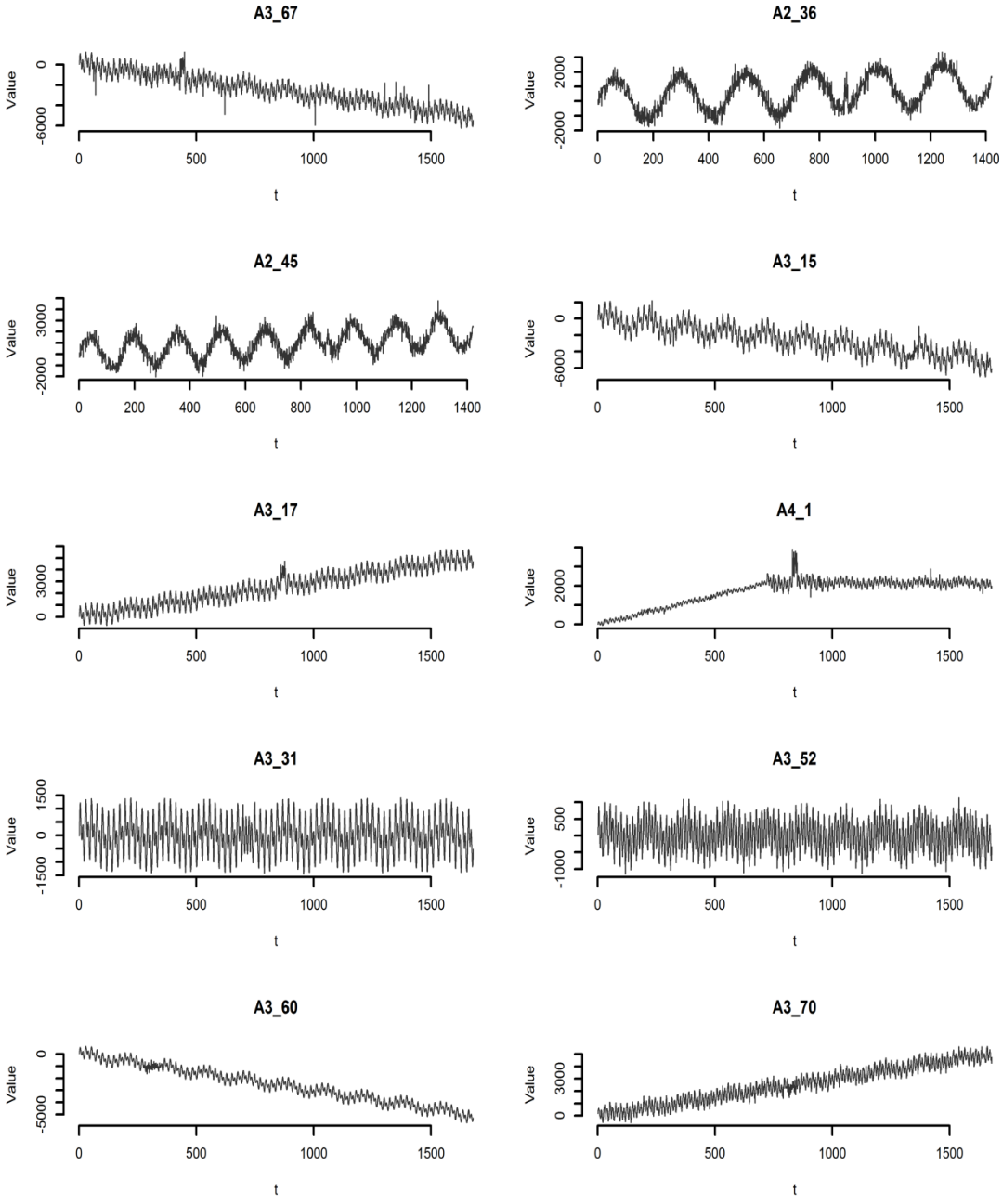


Figure 5.9 Synthetic time series plots

Table 5.8 Parameter settings for synthetic time series application

	Parameters	A3_67	A2_36	A2_45	A3_15	A3_17	A4_1	A3_31	A3_52	A3_60	A3_70
HOT-SAX	Subsequence length	20	12	20	30	20	20	30	20	64	45
	Word length	4	3	4	3	4	5	5	5	8	5
	Alphabet cardinality	5	5	4	7	5	6	6	4	6	5
WAT	Word length	20	12	20	30	20	20	30	20	64	45
	Alphabet cardinality	4	5	5	4	4	6	6	4	5	5
PAV	Precision of the slope	1	1	1	1	2	1-2	1	2	2	1
MPAV	Haar Wavelet Resolution	2	2	2	2	3-4	2-3	2	2	2	2
	Precision of the slope	2	3	1	1	2	1-2	1	2	1	1

Table 5.9 Results for synthetic time series

	A3_67	A2_36	A2_45	A3_15	A3_17	A4_1	A3_31	A3_52	A3_60	A3_70
Anomaly Index	431:459	890:901	890:909	1320:1349	860:879	830:849	700:729	700:719	280:343	801:845
HOT-SAX	441:460 (50.0%)	894:905 **(66.6%)**	893:912 **(85.0%)**	1330:1359 (66.6%)	863:882 **(85.0%)**	843:862 (35.0%)	692:721 (73.3%)	701:720 **(95.0%)**	258:321 (65.6%)	787:831 (68.8%)
	440:459 (50.0%)	895:906 (58.8%)	898:917 (60.0%)		864:883 (80.0%)		699:728 **(96.6%)**		266:329 **(78.1%)**	764:808 (17.7%)
WAT	441:460 (50.0%)	894:905 (66.6%)	881:900 (55.5%)	- (0%)	865:884 (75.0%)	842:861 **(40.0%)**	693:722 (76.6%)	702:721 (90.0%)	- (0%)	772:816 (35.5%)
	444:463 (35.0%)	895:906 (58.8%)	883:902 (65.0%)		869:888 (55.0%)		703:733 (86.6%)			794:838 **(84.4%)**
PAV	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)
MPAV	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)	- (0%)

Ten synthetic time series used in application are given in Figure 5.9 and parameter settings for each time series are given in Table 5.8. Results are given in Table 5.8. The names “A2”, “A3” and “A4” stand for “A2Benchmark”, “A3Benchmark” and “A4Benchmark” in Yahoo Webscope Data respectively. Generally, synthetic time series include point anomalies, therefore point anomalies were transformed into subsequence anomalies with two different ways. First way was changing the seasonality of a sequence which includes anomaly point. Second way was creating an anomaly sequence by using normal distribution with mean of anomaly point and random standard deviations. In general; HOT-SAX found all anomaly sequences and WAT managed to find 9 anomaly sequences out of 10. PAV and MPAV couldn’t find any anomaly sequences, because these algorithms are

highly sensitive to noise. Slope and length values of linear patterns are generally different with high noise; therefore, algorithms consider most of the linear patterns as an anomaly.

50% of the anomaly sequence in time series “A3_17” was revealed by both algorithms HOT-SAX and WAT with same coverage accuracy. “A2_36” had an anomaly sequence length 12, HOT-SAX and WAT found the same sequence with coverage accuracy 66.6%. In time series “A2_45” anomaly sequence formed between 890:909, HOT-SAX had be best coverage accuracy with 85.0%. Anomaly sequence in time series “A3_15” was only found by HOT-SAX, nevertheless coverage accuracy was 66.6%. In time series “A3_17” anomaly sequence was formed with length 20 and HOT-SAX has the best coverage accuracy 85.0%. “A4_1” had a change point and anomaly sequence was formed after change point, HOT-SAX and WAT had relatively lower coverage accuracy than other time series, nevertheless WAT had the best coverage accuracy with %40. Time series “A3_31” had three different seasonality features and anomaly sequence was formed by removing two of the seasonality features of a certain sequence, consequently HOT-SAX and WAT both gave good results and HOT-SAX had the best coverage accuracy with 96.6%. Anomaly sequence in “A3_52” was formed with the same idea (removing some of the seasonality features) and HOT-SAX had the best coverage accuracy with 78.1%. For time series “A3_60” and “A3_70”, HOT-SAX and WAT gave the best results with coverage accuracies 78.1% and 84.4% respectively.

General accuracies for synthetic and real time series are given in Table 5.8. Consequently, HOT-SAX and WAT algorithm performed significantly better than PAV and MPAV algorithms. Parameter settings highly affect the results, in this application algorithms were performed under multiple parameter settings and best results are shown.

Table 5.10 General accuracy of algorithms

	General Accuracy		
	Real time series	Synthetic time series	Average
HOT-SAX	83.76%	72.67%	78.215%
WAT	76.38%	49.87%	63.125%
PAV	21.56%	0.0%	10.78%
MPAV	11.12%	0.0%	5.56%

CHAPTER SIX
CONCLUSION

Temporal data mining is interested in analyzing big amount of temporal data to automatically discover interesting patterns or relationships which gives useful information. Getting the valid, useful information is the main motivation of temporal data mining. In this thesis anomaly detection methods are examined. Different types of anomalies can be observed in temporal data; point anomalies and subsequence anomalies. Detecting subsequence anomalies is an attractive field and several algorithms have been proposed in the literature. In this thesis, Heuristically Ordered Time series using Symbolic Aggregate Approximation (HOT-SAX), Pattern Anomaly Value (PAV), Multi-scale Abnormal Pattern Mining Algorithm (MPAV) and Wavelet and Augmented Trie (WAT) anomaly detection methods were compared by using synthetic and real temporal data sets. Also, different temporal data representation techniques are tested with HOT-SAX algorithm.

In chapter 5.1, representation techniques were tested. Three time series with tagged anomalies were used in the application. HOT-SAX algorithm was preferred in this application because of the structure of representation techniques. SAX, ESAX, TSX and aSAX representation techniques aim to represent time series into symbolic and discrete structure which is appropriate for HOT-SAX algorithm. In general, all representation techniques besides ESAX gave consistent results,

In the second part of the application, anomaly detection techniques were tested. Financial, medical and synthetic time series were used in the application. Synthetic time series were generated from stochastic process with tagged anomaly sequence. All algorithms found the anomaly sequence, however, PAV and MPAV algorithms found additional anomaly sequences.

Financial time series was used in application 5.3. Euro/Dollar exchange rates were used and the time series didn’t have tagged anomaly points. WAT and HOT-SAX algorithms gave almost same results for this application and we can say that there is

an anomalous event occurred between second week of February, 2009 and March, 2009. Also, reason of this anomaly sequence traced back and valid reasons are detailed in 5.3. PAV and MPAV candidate other anomaly sequences, none of the candidates had anything in common, nevertheless these candidates might be anomalies and need to be traced back and proved by economical incident. Electrocardiography data was used in application 5.3. This real time series had tagged anomaly point by a cardiologist. HOT-SAX and WAT algorithms managed to find the anomaly sequence, whereas PAV and MPAV didn't.

In general, trend-based symbolic approximation gave better results than other symbolic representation techniques in terms of anomaly detection. This representation technique was adapted to HOT-SAX algorithm and in this case algorithm requires only two parameters; subsequence length and alphabet cardinality. It should be considered as a considerable pace of ideal parameter free data mining.

HOT-SAX algorithm managed to find anomalies with higher coverage accuracy than other anomaly detection algorithms. With different symbolic representation techniques depending on time series characteristics, this accuracy might be easily get higher. Successes of the all algorithms are highly depending on the parameters, a lot of study has been interested in towards parameter-free data mining and researchers proposing new ideas recently. The methods which require less parameter might be the main motivation of the future works.

REFERENCES

Aggrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. *In: Proceedings of 4th International Conference on Knowledge Discovery and Data Mining***,** 439-450.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). *Time series analysis, forecasting and control*. New Jersey: Englewood Clifs.

Bu, Y., Leung, W., Fu, A. W., Keogh, E., Pei, J., & Meshkin, S. (2007). Wat: Finding top-k discords in time series database. *In Proceedings of 7th SIAM International Conference on Data Mining.*

Chandola, V., Grumbach, A., & Kumar, C. (2009). Anomaly Detection: A survey. *ACM Computing Surveys, 41*(3), Article 15.

Chen, X., & Zhan, Y. (2008). Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics, 214*(1), 227-237.

Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys, 45*(1), Article 12.

European Central Bank Web Site. (2015). Retrieved December 10, 2014, from https://www.ecb.europa.eu/stats.

Fu, T. (2011). A review on time series data mining, *Engineering Applications of Artificial Intelligence*, *24*, 164-181.

Fu, A. W., Leung, W., Keogh, E., & Lin, J. (2006). Finding time series discords based on haar transform. *In Proceeding of the 2nd International Conference on Advanced Data Mining and Applications. Springer Verlag, 4093,* 31-41.

Gupta, M., Gao, J., Aggarwal, C. C., & Han, J. (2013). Outlier detection for temporal data: a survey. *IEEE Transactions on Knowledge & Data Engineering, 26,* 2250-2267.

Jakkula, V., Cook, D.J., & Crandall, A. (2007). Temporal pattern discovery for anomaly detection in a smart home. *Proceedings of the 3rd IET Conference on Intelligent Environments,* 339-345.

Keogh, E. (1997). Fast similarity search in the presence of longitudinal scaling in time series databases. *Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence*. 578–584.

Keogh, E., & Pazzani, M. (2000). A Simple dimensionality reduction technique for fast similarity search in large time series databases. *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining.* 122–133.

Keogh, E. (2008). Indexing and mining time series data. *Enclopedia of GIS*, 493-497.

Keogh, E., Lin, J., Lee, S.H., & Herle, H.V. (2005). Finding the most unusual time series subsequence: algorithms and applications, *Knowledge and Information Systems, 11*(1), 1-27.

Keogh, E., Lin, J., & Fu, A. (2005). HOT-SAX: Efficiently finding the most unusual time series subsequence. *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, 226-233.

Keogh, E., Zhu, Q., Hu, B., Hao. Y., Xi, X., Wei, L. et al. (2011). *The UCR Time Series Classification / Clustering Homepage:* http://www.cs.ucr.edu/~eamonn/time_series_data/.

Laxman, S., & Sastry, P. S. (2006). A survey of temporal data mining. *Sadhana, 31*(2), 173-198.

Li, G., Zhang, L., & Yang, L. (2012). TSX: a novel symbolic representation for financial time series, *PRICAI 2012: Trends in Artificial Intelligence Lecture Notes in Computer Science, 7458*, 262-273.

Lin, J., Keogh, E., Leonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. 2-11.

Lkhagva, B., Suzuki, Y., & Kawagoe, K. (2006). New time series data representation ESAX for financial applications. *Proceedings of the 22nd International Conference on Data Engineering Workshops*. 17–22.

Ma, J., & Perkins, S. (2003). Online novelty detection on temporal sequences. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 613–618.

Meyer, S.C. (2005). Analysis of base flow trends in urban streams, northeastern Illinois, USA. *Hydrogeol. Journal*, *13*, 871–885.

Mitsa, T. (2010). *Temporal Data Mining*. CRC Press, Boca Raton, Florida.

Onat, M. Y., Yıldıztepe, E. (2015). Zamansal veri madenciliği ve anomali tespiti için bir uygulama. *Akademik Bilişim, Subat, 2015 – Eskişehir*.

Pham, N. D., Quang, L. L., & Dang, T. K. (2010). HOT aSAX: A novel adaptive symbolic representation for time series discords discovery. *Information and Database Systems Lecture Notes in Computer Science, 5990*, 113-121.

Ratanamahatana, C. A., Lin, J., Gunopulos, D., & Keogh, E. (2005). Mining time series data. *Data Mining and Knowledge Discovery Handbook*, 1069-1103.

Tufte, E. R. (1983). *The visual display of quantitative information*. Chesire, CT: Graphics Press.

Türkiye Bankalar Birliği, Bankacılık ve Araştırma Grubu. (2009). Uluslararası finansal kriz karşısında ülkeler tarafından alınan önlemler kronolojisi. *Bankacılar Dergisi*, *69*, 86-93.

Yahoo Webscope. (2015). *S5 - A Labeled Anomaly Detection Dataset, version 1.0.* Retrieved June 20, 2015 from http://webscope.sandbox.yahoo.com/.

Yi, B., & Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. *Proceedings of the 26th International Conference on Very Large Data Bases*. 385–394.

Wang, R. (2013). *Haar transform*. Retrieved May 10, 2015, from http://fourier.eng.hmc.edu/e161/lectures/Haar/haar.html.

APPENDICES

#SAX

```
SAX<-function(x,w) {
  sax<-character()
  bpoints<-c()
  for (i in 1:(w-1)) { bpoints<-rbind(bpoints,qnorm(i/w))}
  ltrs<-letters[1:(w)]
  for (i in 1:length(x)){
    k<-w-1
    repeat{
      if (sum(x[i]<bpoints)==k) {
        sax[i]<-ltrs[w-k]
        break }
      k<-k-1}
  }
  return(sax)}
```

#PAA

```
paa<-function(x,y){
  paa<-c()
  for (i in 1:nrow(x)){ paa<-rbind(paa,matrix(x[i,],ncol=y,byrow=T))}
return(paa)}
```

#All possible subsequences

```
sseq<-function(x,y){
  ss<-c()
  o.ss<-c()
  for (i in 1:y) {
    if(i==1){ss<-rbind(ss,matrix(x[i:length(x)],ncol=y,byrow=T)) }
    else {ss<-rbind(ss,matrix(x[i:(length(x)-(y-i+1))],ncol=y,byrow=T))}}
  return (ss)}
```

#HOT-SAX

```
#x=Time series, y=subsequence length, z=alphabet cardinality, p=word length
hotsax<-function(x,y,z,p){
  length(x)<-length(x)-length(x)%%y
  aldist<-c()
  nndist<-c()
  s<-sseq(scale(x),y)
  s.paa<-paa(s,y/p)
  ptrn<-matrix(SAX(apply(s.paa,1,mean),z),ncol=p,byrow=T)
  ptrn<-apply(ptrn,1,paste,collapse="")
  s.ptrn<-sort(table(ptrn))
  ind<-which(ptrn%in%names(s.ptrn[s.ptrn==1]))
  t.ind<-sseq(c(1:length(x)),y)
  for (i in ind) {
```

```
   for (j in 1:nrow(s)) {if (abs(t.ind[i,1]-t.ind[j,1])>=y) {
      aldist<-rbind(aldist,c(dist(rbind(s[i,],s[j,])))) }}
   nndist<-rbind(nndist,min(aldist))}
result<data.frame(Pattern=ptrn[ind],Start=t.ind[ind,1],End=t.ind[ind,y],NN=nndist,in
d)
 return(result)  }
```

#WAT

```
require(wavelets)
wat<-function(x,y,z){
 length(x)<-length(x)-length(x)%%y
 s<-sseq(x,y)
 w<-dwt(s[1,],filter="haar")
 k<-w@level
 repeat{
  allhaar<-c()
   for (i in 1:nrow(s)){
    w<-dwt(s[i,],filter="haar")
    allhaar<-rbind(allhaar,w@V[[k]])}
  allhaar<-scale(allhaar)
  sr<-matrix(SAX(allhaar,z),ncol=length(w@V[[k]]),byrow=T)
  sr<-apply(sr,1,paste,collapse="")
  k<-k-1
  print(k)
  if (min(table(sr))==1) break }
 s.sr<-sort(table(sr))
 ind<-which(sr%in%names(s.sr[s.sr==1]))
 t.ind<-sseq(c(1:length(x)),y)
 aldist<-c()
 nndist<-c()
 for (i in ind) {
  for (j in 1:nrow(s)) {
   if (abs(t.ind[i,1]-t.ind[j,1])>=y) {
    aldist<-rbind(aldist,dist(rbind(s[i,],s[j,])))}
  }
  nndist<-rbind(nndist,min(aldist))
 }
 result<-
datata.frame(Pattern=sr[ind],Start=t.ind[ind,1],End=t.ind[ind,y],NN=nndist,ind)
return(result)  }
```

#PAV

```
require(wavelets)

#Features of linear patterns (a=time series, e=precision of slope)
features<-function(a,e) {
  f<-data.frame()
  for (i in 1:(length(a)-1)){
    s<-round((a[i+1]-a[i]),e)
    l<-sqrt((a[i+1]-a[i])^2+1)
    f<-rbind(f,cbind(s,l))
  }
return (f)}

#Support Count
sc<-function(x){
  scount<-c()
    for(i in 1:nrow(x)){
    a<-table(x$s[i]==x$s & x$l[i]==x$l)
    scount<-rbind(scount,a[2])
  }
return (scount)
}

#min-max norm - Anomaly Value
av<-function(x){
  n.head<-(x-min(x))/(max(x)-min(x))
  av<-1-n.head
  return(av)
}

#Pav
pav<-function(x,e){
  a<-features(x,e)
  b<-sc(a)
  s<-av(b)
  plot.ts(s)
 return(which(max(s)==s))}
```

#MPAV

```
# x=time series, k=Haar wavelet resouliton, z=PAV parameter
mpav<-function(x,k,z){
  w<-dwt(x,filter="haar")
  a<-features(w@V[[k]],z)
  b<-sc(a)
  s<-av(b)
  plot.ts(s)
  return(which(max(s)==s)*2^k)
}
```