การตรวจหาข้อมูลที่ผิดปกติบนอนุกรมเวลาจากหน้าต่างอนุกรมย่อยเพื่อนบ้านไกลสุด

นายเสนีย์ กิติมูล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

ANOMALY DETECTION ON TIME SERIES FROM FURTHEST NEIGHBOR WINDOW SUBSERIES

Mr. Senee Kitimoon

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Applied Mathematics and Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2016

Thesis Title	ANOMALY DETECTION ON TIME SERIES FROM FURTHEST NEIGHBOR WINDOW SUBSERIES
By	Mr.Senee Kitimoon
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Assistant Professor Krung Sinapiromsaran, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

..... Dean of the Faculty of Science
(Associate Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Boonyarit Intiyot, Ph.D)

..... Thesis Advisor
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

..... Examiner
(Jiraphan Suntornchost, Ph.D.)

..... External Examiner
(Assistant Professor Chumphol Bunkhumpornpat, Ph.D.)

เสนีย์ กิติมูล : การตรวจหาข้อมูลที่ผิดปกติบนอนุกรมเวลาจากหน้าต่างอนุกรมย่อยเพื่อนบ้านไกลสุด. (ANOMALY DETECTION ON TIME SERIES FROM FURTHEST NEIGHBOR WINDOW SUBSERIES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.กรุง สินอภิรมย์สราญ, 53 หน้า.

การตรวจหาข้อมูลที่ผิดปกติบนอนุกรมเวลา แบ่งได้เป็นสามประเภท คือ ความผิดปกติแบบจุด ความผิดปกติเมื่อเทียบกับบริเวณข้างเคียง และความผิดปกติเมื่อรวมกันเป็นกลุ่ม งานวิจัยนี้ นำเสนอวิธีการตรวจจับความผิดปกติบนข้อมูลประเภทอนุกรมเวลา เรียกว่า การตรวจหาข้อมูลที่ผิดปกติบนอนุกรมเวลาจากหน้าต่างอนุกรมย่อยเพื่อนบ้านไกลสุด ค่าควอร์ไทล์ทั้งสามค่าซึ่งถูกใช้เป็นตัวแทนการแจกแจงจะถูกคำนวณและหักออกด้วยข้อมูลตัวแรก ในหน้าต่างอนุกรมเวลานั้น เวกเตอร์ของควอร์ไทล์ทั้งสามค่า ได้แก่ ควอร์ไทล์บน มัธยฐาน และควอร์ไทล์ล่าง จะถูกใช้เพื่อการคำนวณหาค่าระยะทางระหว่างหน้าต่างย่อย และหาระยะทางไปถึงเพื่อนบ้านตัวที่ k เพื่อนำมาใช้เป็นค่าคะแนน กลุ่มของคะแนนมิติเดียว จะถูกเรียงเพื่อคำนวณหาค่าควอร์ไทล์ เกณฑ์พิสัยควอร์ไทล์จาก บอกซ์พลอตที่ถูกปรับสำหรับการกระจายเบ้ถูกนำมาใช้เพื่อระบุจุดผิดปกติ การทดลองบนชุดข้อมูลอนุกรมเวลาที่ใช้มาจาก เบนช์มาร์กของยาฮูถูกใช้และประเมินผลด้วยตัววัด พรีซีชัน, รีคอลล์ และ เอฟ-เมเชอร์ ผลที่ได้แสดงให้เห็นว่า เอฟเอ็นดับเบิ้ลยูเอส มีประสิทธิภาพและมีความแม่นยำมากกว่า 80% ในทุกๆ ตัววัด

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต .....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
ปีการศึกษา	2559	

5772255623: MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORDS: ANOMALY DETECTION / TIME SERIES / FURTHEST NEIGHBOR WINDOW SUBSERIES / CONTEXTUAL ANOMALY

SENEE KITIMOON : ANOMALY DETECTION ON TIME SERIES FROM FURTHEST NEIGHBOR WINDOW SUBSERIES. ADVISOR : ASST. PROF. KRUNG SINAPIROMSARAN, Ph.D., 53 pp.

Anomaly detection in time series is classified into three types which are point anomaly, contextual anomaly, and collective anomaly. This work proposes a novel method called the Furthest Neighbor Window Subseries (FNWS) for detecting contextual anomalies which normally appear in a time series dataset. Three quartiles representing a local distribution are computed and relocated by subtracting the first data point in the window subseries. A vector of three quartiles —the lower quartile, the median and the upper quartile —is used to compute the distances among all window subseries and the furthest k-nearest neighbor distance is picked as the score. The collection of the one-dimensional score is sorted and the score quartiles are computed. The interquartile range rule from the adjusted boxplot for skew distributions is applied to identify anomalies. The empirical experiments on the benchmark time series datasets from Yahoo with a list of labeled outliers are performed and evaluated using precision, recall, and F-measure. The results show that FNWS works effectively and accurately having the average scores more than 80% on all metrics.

Department	: Mathematics and Computer Science	Student's Signature	.....
		Advisor's Signature	.....
Field of Study	: Applied Mathematics and Computational Science		
Academic Year	: 2016		

Acknowledgements

First and foremost, I would like to thank my advisor, Assistant Professor Dr. Krung Sinapiromsaran, for his valuable support and help throughout the course of my Masters work. I would like to thank the Applied Mathematics and Computational Science at the department of Mathematics and Computer Science at Chulalongkorn University for providing me with the excellent facilities during my graduate studies.

In addition, I would like to acknowledge all my dissertation committee members: Assistant Professor Dr. Khamron Mekchay, Dr. Jiraphan Suntornchost, and Assistant Professor Dr. Chumphol Bunkhumpornpat, for their contribution and the advice in the completion of my dissertation.

Moreover, I would like to acknowledge the Development and Promotion of Science and Technology Talented (DPST) project for providing a financial support throughout my study.

Finally, I would like to thank all members of our Data Mining Group for providing a friendly environment and suggestion on my work. I would also like to thank all my family members and friends and colleagues in the AMCS program who have constantly motivated me all through my work.

Contents

Chapter Page

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

Data mining or knowledge discovery in databases (KDD) is a process of finding or extracting useful information from a large database and transforms it into a meaningful structure for further use. Data mining tasks can be divided into six different categories: classification, regression, clustering, summarization, dependency modeling, and deviation detection. [11]

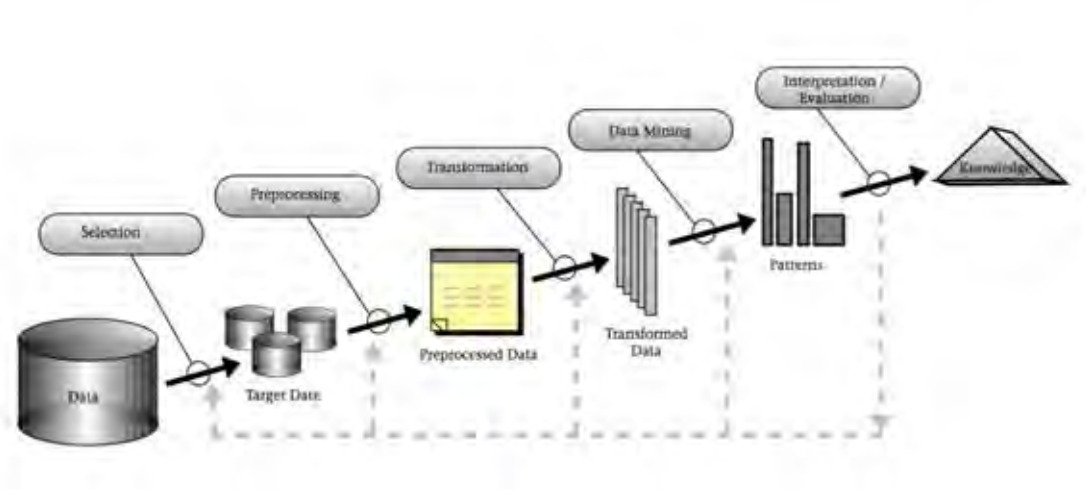


Figure 1.1: Knowledge discovery in databases process [11]

Anomaly detection or outlier detection is one of the data mining tasks in deviation detection category. It is defined as a problem of finding an individual object that behaves very differently from normal (majority) objects. This anomalous might be an interesting data point or an error that requires further investigation by an expert.

In many domains, the data tend to be collected with time stamps, such as heartbeat pulse from the electrocardiogram data, patient's respiration, number of

tweets per second of Twitter, CPU usages of the computer servers, ocean tides, the daily closing values of the stock markets and the flight data collected in the form of sequences of observations from various aircraft sensors during the flight. This specific kind of data is classified as the time series data. One important property that all time series data share in common is that it is ordered. If a user swaps a data point from one location to another, then the interpretation may change. This property makes time series differ from the ordinary static data.

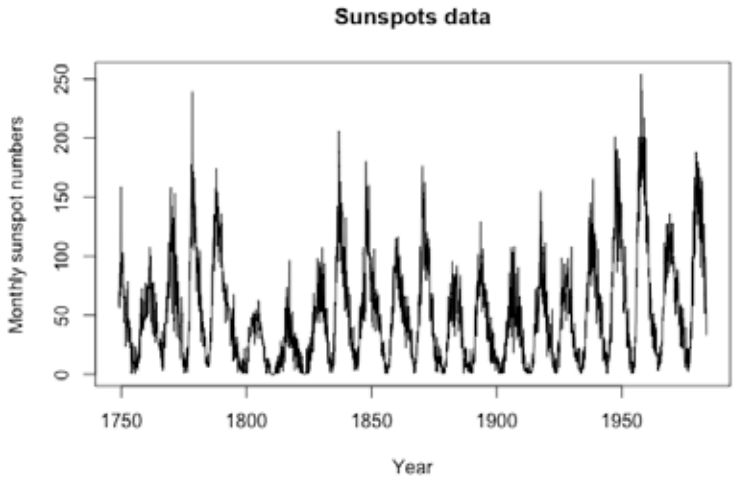


Figure 1.2: Monthly average of relative sunspot numbers from 1749 to 1983. The data is collected by Swiss Federal Observatory and Tokyo Astronomical Observatory. [2]

The applications of anomaly detection appear in various fields such as detecting anomalous heartbeat pulse from the electrocardiogram data [12], detecting the number of unusual tweets per second of Twitter [21], detecting anomalous CPU usages of the computer servers, detecting anomalous patient's respiration [17]. The purpose of anomaly detection has two aspects. First, anomaly detection can warn a responsible person of the abnormal behavior, for example, detecting anomalous heart beat pulse could early warn a doctor about patient's heart disease. Second, it is used for identifying a data point that is needed to be cleaned which alleviates a predictive model to perform better especially the method that is vulnerable to

anomalous data such as regression.

A primary goal of time series analysis is forecasting. Many techniques on time series analysis such as ARIMA, VAR [22] involve building a model for forecasting which is vulnerable to anomalous data. So, detecting anomalous data and managing it before performing any time series analysis will help make the result more reliable.

This thesis proposes an automatic anomaly detection in time series, the Furthest Neighbor Window Subseries (FNWS). The algorithm requires two parameters n and k where n represents the size of the window subseries where each window subseries is called an element. The distribution for each element is represented by three numbers, the lower quartile, the median and the upper quartile subtracting the first data point in the window. The Euclidean distances among all elements of three numbers are computed, and the k-nearest neighbor distance of each element is extracted to represent the score. The anomaly criterion was adapted from adjusted boxplot [16] to identify anomalous data points.

1.1 Objectives

The objective of this thesis is to propose a novel method for detecting contextual anomaly on time series data and compare with other methods and analyze it based on synthesized datasets and benchmark datasets.

1.2 Scope of Work

The Furthest Neighbor Window Subseries is an unsupervised algorithm which needs no label of data. The considered data are real-valued univariate time series data. The method requires a measure of dissimilarity which is evaluated via the distance metric. All implementation and empirical experiments are performed

on Python programming environment using Jupyter notebook. The results of detection are evaluated using three measurement metrics: precision, recall, and F-measure.

1.3 **Expected Outcome**

The 100 time series data from Yahoo with various trends and seasons are used in the experiments to test FNWS. The results from the FNWS method is expected to achieve the best or higher score on most time series data on precision, recall, and F-measure metric comparing with other methods.

1.4 **Thesis Overview**

This thesis is organized as follows. Difference technique for preprocessing time series and existing method for anomaly detection are presented in Chapter 2. The new method proposed in this thesis and its implementation are introduced in Chapter 3. The empirical experiments and results are analyzed in Chapter 4. Lastly, the conclusion and discussion are drawn in Chapter 5.

CHAPTER II

BACKGROUND KNOWLEDGE

This chapter introduces the preliminary knowledge of this work. There are two main sections in this chapter: basic statistical knowledge and anomaly detection on time series. Prior work on anomaly detection method for univariate time series will also be discussed in this chapter.

2.1 Statistics

Statistics is a subset of mathematics used to describe a behavior and relationships of data. This section describes a basic statistical method that is used in defining central tendency and dispersion of data. Different ways of displaying data are also explained in this section.

2.1.1 Measures of Center

A measure of center or average is a number expressing the central of data. The simplest and the most widely used in measuring central tendency is the arithmetic mean which will be described in this section. Other popular types of average are median and harmonic mean.

2.1.1.1 Mean

The mean (sometimes can be called average), or the arithmetic mean, is the quantity that commonly used to describe central tendency of a set of real values. The mean of a data set can be calculated by adding up all values in the data set and dividing it by the number of values in that data set. Mathematically, the

arithmetic mean of the real number $x_1, x_2, \ldots, x_n$, is defined to be

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{2.1}$$

Example 2.1**.** Consider the following set of values: $A_1 = \{1, 2, 3, 4, 5\}$. The arithmetic mean of this set would be:

$$\bar{x} = \frac{1 + 2 + 3 + 4 + 5}{5} = 3.$$

Another example with one value deviates from the others: $A_2 = \{1, 2, 3, 4, 100\}$. The arithmetic mean of this set would be:

$$\bar{x} = \frac{1 + 2 + 3 + 4 + 100}{5} = 22.$$

We can see that the arithmetic mean is sensitive to outliers in the data.

2.1.1.2 **Median**

The median is the middle value in a real-valued data set. It is the value in the center if a data set has been ordered. Given ordered set A of real numbers $x_1, x_2, \ldots, x_n$, the median of set A is defined by

$$\operatorname{med} A = \begin{cases} x_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{x_{n/2} + x_{n/2+1}}{2} & \text{if } n \text{ is even} \end{cases}. \tag{2.2}$$

Example 2.2**.** With the same set of examples as above. Since both set A_1 and A_2 are already ordered, the median of both sets are

$$\operatorname{med} A_1 = \operatorname{med} A_2 = x_3 = 3.$$

In contrast with mean, median seems not to be effected from outliers in the data.

2.1.1.3 **Harmonic Mean**

The Harmonic mean is a special type of mean. It usually used to average the particular type of values like speed, rate, and ratio. For example, a car is driven from town A to town B, which is 120 km away, with the speed of 40 km per hour and is returned with the speed of 120 km per hour. If the average speed is normally calculated, it would be $\frac{40+120}{2} = 80$ km per hour which may not true capture the time spend on each trip. Thus, the appropriate average should be the total distance divided by total time: $\frac{2*120}{4} = 60$ km per hour. Note that, the harmonic mean can be used to calculate this kind of mean. For the real numbers $x_1, x_2, \ldots, x_n$, the harmonic mean can be defined by

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}. \tag{2.3}$$

Example 2.3**.** Averaging speed of 2 trips, 40 and 120 km per hour, with the same distance can be calculated as

$$H = \frac{2}{\frac{1}{40} + \frac{1}{120}} = 60.$$

2.1.2 **Measures of dispersion**

Measuring a dispersion is used to show how much a data spread. It can help to express the approximate distribution of the data.

2.1.2.1 **Range of the Data**

The range of the data can be interpreted as the maximum possible difference in the data. Given a set of real values X, the range is simply defined by

$$\text{range}(X) = \max X - \min X. \tag{2.4}$$

Example 2.4**.** Let X be a set of values, $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$, the range of set X can be calculated as

$$\text{range}(X) = 11 - 1 = 10.$$

2.1.2.2 **Standard Deviation**

The Standard Deviation (SD) is a value that is used to determine the spread of a distribution of data respect to the mean. A low value of standard deviation (close to zero) shows that the data tend to stay close to the mean. In contrast, a high value of standard deviation shows that the data deviate far from the mean. The standard deviation is defined to be

$$s = \left(\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1} \right)^{\frac{1}{2}}. \tag{2.5}$$

Example 2.5**.** Let X be the same set as in example 2.4, standard deviation of set X can be calculated as

$$\bar{x} = \frac{1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11}{11} = 6$$

$$s = \left(\frac{\sum_{i=1}^{11} (i - 6)^2}{11 - 1} \right)^{\frac{1}{2}} = \sqrt{11} \approx 3.32.$$

2.1.2.3 Quartiles and Interquartile Range

The quartiles are the values that divide data set into four equal-sized subsets, and each subset contains a quarter of the data set: the lowest 25% of numbers, the next lowest 25% of numbers (up to the median), the second highest 25% of numbers (above the median), and the highest 25% of numbers. The values that divide these part are denoted by Q_1 (the lower quartile), Q_2 (the median), and Q_3 (the upper quartile) respectively.

The interquartile range (IQR), or sometimes called midspread, is the measure of dispersion that evaluates the range of middle 50% of the data set. The formula of IQR is defined to be

$$IQR = Q_3 - Q_1. \tag{2.6}$$

Example 2.6**.** Let X be the same set as in example 2.4, the IQR of set X can be calculated as

$$IQR = 8.5 - 3.5 = 5.$$

2.1.3 Measure of Skewness

The measure of skewness explains asymmetry of the data. The value can be positive or negative, or even undefined. This work uses medcouple [5] as a measure of skewness.

2.1.3.1 Medcouple

The medcouple is one of robust skewness measures operating on univariate data. It is used in adjusted box plot which performs very well when data have outliers.

Given an ordered set of real values $X = \{x_1, x_2, \ldots, x_n\}$, that is $x_1 \leq x_2 \leq$

$\cdots \leq x_n$, the *medcouple* of X is defined by:

$$MC(X) = \underset{x_i \leq Q_2 \leq x_j}{\text{med}} h(x_i, x_j) \tag{2.7}$$

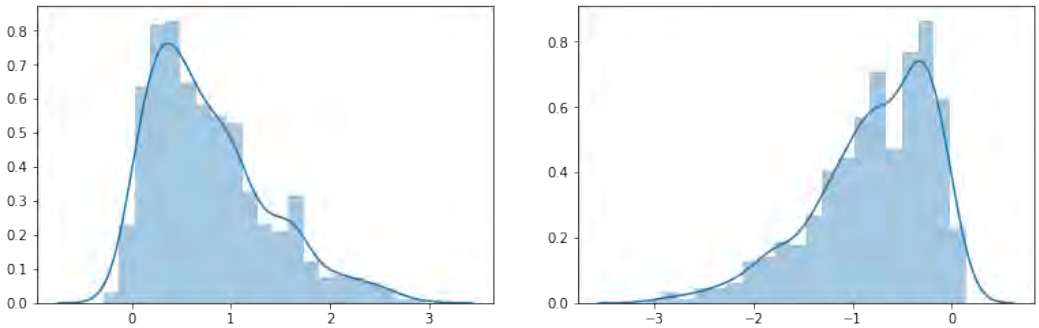
where Q_2 is the median of X and function h is given by:

$$h(x_i, x_j) = \frac{(x_j - Q_2) - (Q_2 - x_i)}{x_j - x_i}. \tag{2.8}$$

The special case $x_i = Q_2 = x_j$ takes the value of the from $sgn(n-1-i-j)$ instead, where sgn is the sign function.

Example 2.7**.** Example of different sets and their medcouples:

- $A_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $MC = 0$
- $A_2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100\}$, $MC = 0$
- $A_3 = \{1, 2, 2, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10\}$, $MC = 0.5$
- $A_4 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9, 9, 10, 10\}$, $MC = -0.375$



(a) The positive skew distribution with $MC = 0.2189$

(b) The negative skew distribution with $MC = -0.1232$

Figure 2.1: Examples of skew distrubutions with MC values

2.1.4 Visualizing Data

2.1.4.1 Scatter Plot

The scatter plot is a graph to visualize of two or three variables along two or three axes, respectively. The pattern of the resulting points reveals a correlation in the data set. It is the simplest way to visualize a data set having small number of dimensions.

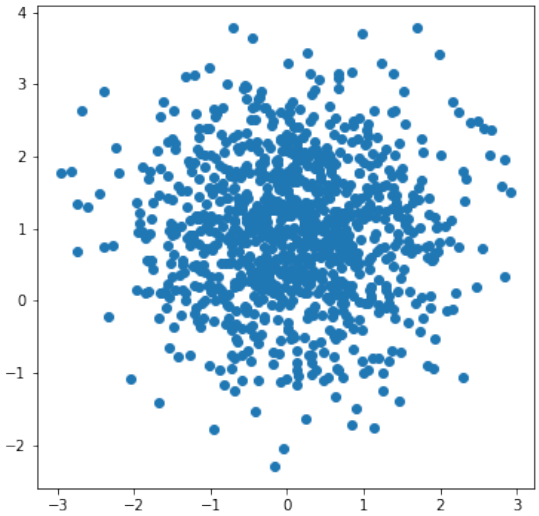


Figure 2.2: An example of a scatter plot of 1,000 sample. Each variable was drawn from normal distribution with mean 0 and sd 1.

2.1.4.2 Box Plot

The box plot (also called box and whisker diagram) shows the rough distribution of an univariate data. A box plot for a data set can be created based on the five statistics: minimum, lower quartile (Q_1), median (Q_2), upper quartile (Q_3), and maximum. The simplest box plot is constructed by two rectangles span from a median line (usually lay horizontally) to the first quartile and the third quartile, this box represents the IQR of the data set. Then the box has lines extending vertically from it (whiskers) covering the data that fall between $Q_2 - 1.5 * IQR$

to $Q_3 + 1.5 * IQR$. Any data that fall outside this range is plotted as an outlier with a dot or small circle. Normally, the mean is also computed and places in the box plot.

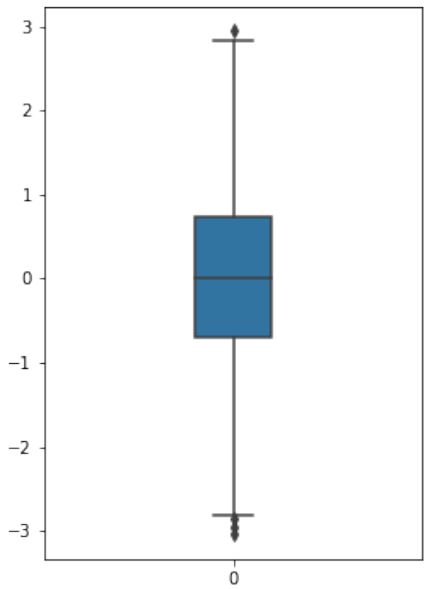


Figure 2.3: An example of a box plot of 1,000 sample drawn from normal distribution with mean 0 and sd 1.

2.2 Time Series

A time series data will be defined as a m-dimensional vector. Since time series is an ordered set of an infinite horizon, but only a finite number of data points can be observed, hence it can be defined as a *Restricted Time Series* as follow:

Definition 2.1 (Restricted Time Series)**.** A restricted time series $T = (t_1, t_2, \ldots, t_m)$ is an ordered set of m real-valued variables.

2.2.1 Type of Anomaly on Time Series Data

Anomalous data on time series can be divided into three types [7]: point anomaly, contextual anomaly, and collective anomaly.

2.2.1.1 Point Anomaly

Point anomaly is the first and obvious type of anomaly. An individual point is considered to be anomalous if it deviates very far from other points in the rest of the time series data. An anomaly of this type is the easiest and simplest type to detect. Many state-of-the-art techniques for static data can be applied, for example, LOF [4] and OOF [6].

2.2.1.2 Contextual Anomaly

In this type, anomalies are the individual instances of the time series in a specific context surrounding it, so it is anomalous in that context but it may not be anomaly with respect to the whole data. In some literature, the contextual anomaly is also referred to as conditional anomaly.

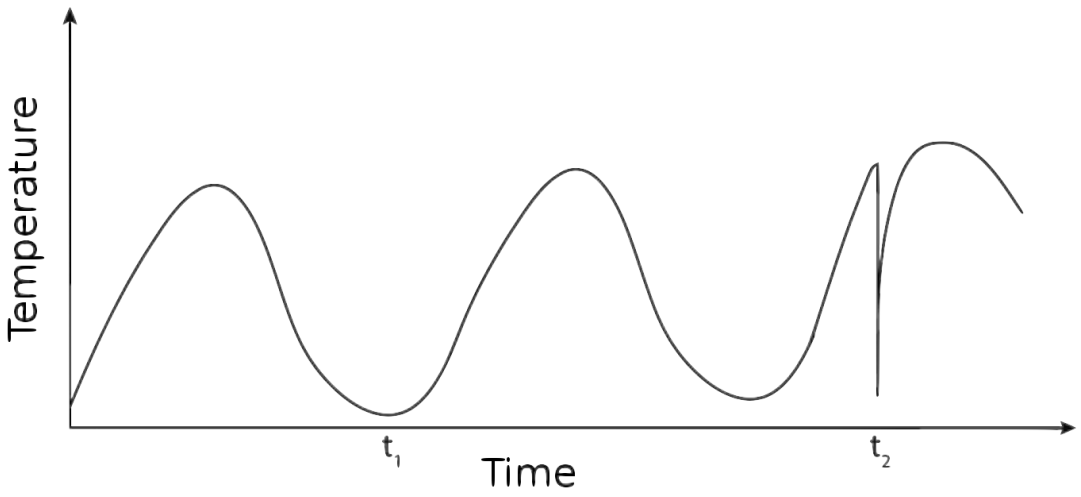


Figure 2.4: An example of a contextual anomaly in time series data [7]

Contextual anomalies have been one of the most commonly investigated in time series data. Figure 2.4 illustrates one such example of a contextual anomaly in time series data. As the example shows, the temperature at time t_1 is equal to that temperature at time t_2 but it occurs in a different context hence the temperature at time t_1 is not considered as an anomaly.

2.2.1.3 **Collective Anomaly**

Collective anomalies are a collection of contiguous data points that appear to be anomalous respect to the entire dataset. The individual data point in a collective anomaly may not be anomalies by itself, but its occurrence together as a collection is anomalous.

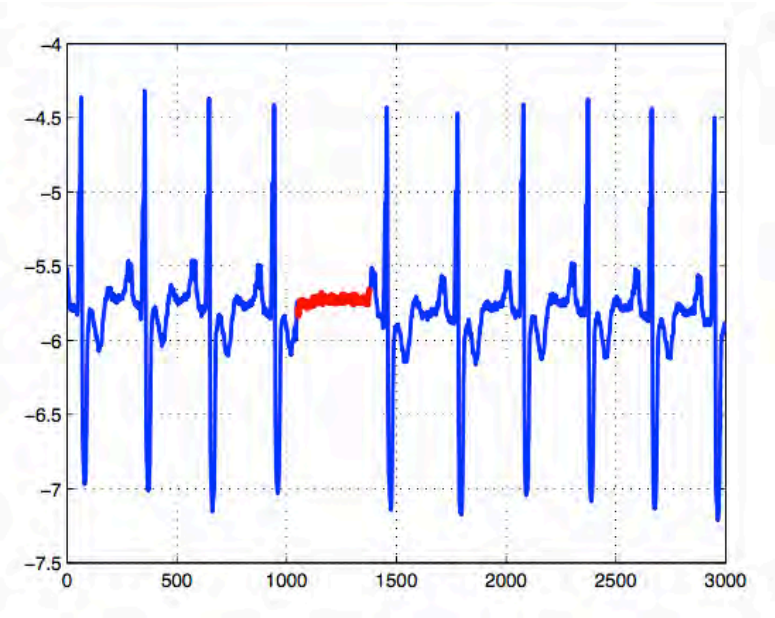


Figure 2.5: An example of a contextual anomaly in a human electrocardiogram data. Values in the range [5000, 7000] represent a collective outlier because the same low value exists for an abnormally long time. [7]

2.2.2 Anomaly Detection Approach

Anomaly detection approach highly depends on the availability of labels in a training data set. Each data point in a dataset is associated with its label which describes whether that instance is normal or anomalous.

Based on the extent to which the labels are available, anomaly detection techniques can operate in one of the following three modes:

2.2.2.1 Supervised Anomaly Detection

The supervised anomaly detection technique builds a model on a dataset where each instance is labeled as "normal" or "anomalous." A typical approach in such cases is to construct a predictive model for normal against anomaly class. The supervised anomaly detection technique has two main problems. First, the anomalous instances are far fewer compared to the normal instances in the training data. This issue is addressed as an imbalance class problem. Second, obtaining the label for data instance accurately is usually hard; it is essentially done by human, but if more data are available (in big data problem), this is impractical.

2.2.2.2 Semi-Supervised Anomaly Detection

The semi-supervised anomaly detection technique operates on a dataset where each instance is labeled only "normal" and builds a model based on the normal behavior of a dataset. A data point that does not agree with the model will be identified as an outlier. Since this technique does not require the anomalous label in training stage, it is more widely applicable than supervised techniques, especially, in a problem that the behavior of anomalous data is unknown.

2.2.2.3 Unsupervised Anomaly Detection

The unsupervised anomaly detection technique is the most widely studied due to the absence of labeled datasets. This technique assumes the majority of data points in a dataset is normal and an instance that behaves differently from the majorities will be identified as an outlier. This technique also makes an assumption that normal instances are far more frequent than anomalies in both the training and testing data.

For a static dataset, there are many methods and algorithms that work effectively such as Local Outlier Factor (LOF) [4], Connectivity-based Outlier Factor (COF) [**Tang2002**], Histogram-based Outlier Score (HBOS) [13]. While these methods work quite well with the anomaly of the first type, which is the point anomaly; they, however, have a hard time extend to detect anomalous data point in a time series which involves temporal characteristic.

2.2.3 Output of Anomaly Detection

One important thing about anomaly detection is how it reports the results. Typically, the results produced by anomaly detection techniques are one of the following two types:

2.2.3.1 Labels

Results of this type will label the data as “normal” or “anomalous” to each data instance.

2.2.3.2 Scores

Results of this type use numerical value to represent the anomaly of each data instance. The high score is usually interpreted as anomalous while the low score

is treated as normal. Some approaches report it as a probability of anomaly. This approach can also be converted to labeling using a cut-off threshold for selecting top few anomalies for further analysis.

2.2.4 **Processing Time series Data**

There are many ways to handle time series data. One of the most popular techniques using for manipulating time series is sliding windows technique.

2.2.4.1 **Sliding Window**

Sliding window is a technique to process a time series and transforms it into *Window Subseries*. A window subseries of a given time series can be defined as follows:

Definition 2.2 (Window Subseries)**.** Given a time series T of length m, a window subseries S_i of T is a contiguous data points of length $n \leq m$ from T starting at index i, that is, $S_i = (t_i, t_{i+1}, \ldots, t_{i+n-1})$ for $1 \leq i \leq m - n + 1$

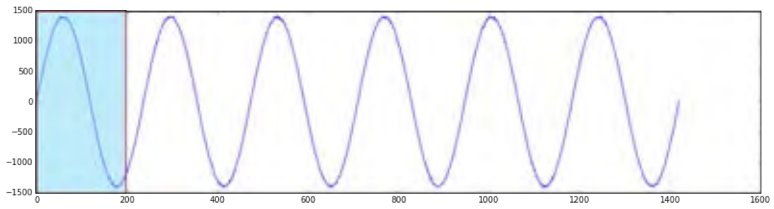


Figure 2.6: Window Subseries of length 200 starting at index 1

Note that there are many variations of sliding window techniques, some of them using *step_size* as a parameter to make the window subseries shift more than just one time step.

2.2.4.2 Distance Measure

In this section, three different distance measures will be investigated. They use for evaluating similarity/dissimilarity of data which are Euclidean distance, Hamming distance, and Dynamic Time Warping distance.

2.2.4.3 Euclidean Distance

Euclidean distance is the most popular distance measure that has been used in wide range areas including in time series data.

Definition 2.3 (Euclidean Distance)**.** Given two time series of the same length $T = (t_1, t_2, \ldots, t_m)$ and $S = (s_1, s_2, \ldots, s_m)$. The Euclidean distance between T and S is defined by

$$\text{EuclideanDist}(T, S) = \sqrt{\sum_{i=1}^{m}(t_i - s_i)^2} \tag{2.9}$$

2.2.4.4 K-Nearest Neighbors Distance

K-nearest neighbors algorithm is best known as a method for classification [18] and regression [1] in the machine learning field. This work applies k-nearest neighbors algorithm and uses it as a score for each window subseries.

Definition 2.4 (K-Nearest Neighbors)**.** Let D be a set of vectors. Given a vector $v \in D$, $K_v \subseteq D$ is the set of k-nearest neighbors of v, if the following conditions hold:

- There are at least k vectors u' in set K_v such that $d(v, u') \leq d(v, u)$ for all u in set $D \setminus K_v$.
- There are at most k vectors u' in set K_v such that $d(v, u') < d(v, u)$ for all

u in set $D \setminus K_v$.

K-nearest neighbors distance is defined as the largest distance from v to all points in K_v which can be written as follows:

$$knnDist(v) = \max_{u \in K_v} d(v, u) \tag{2.10}$$

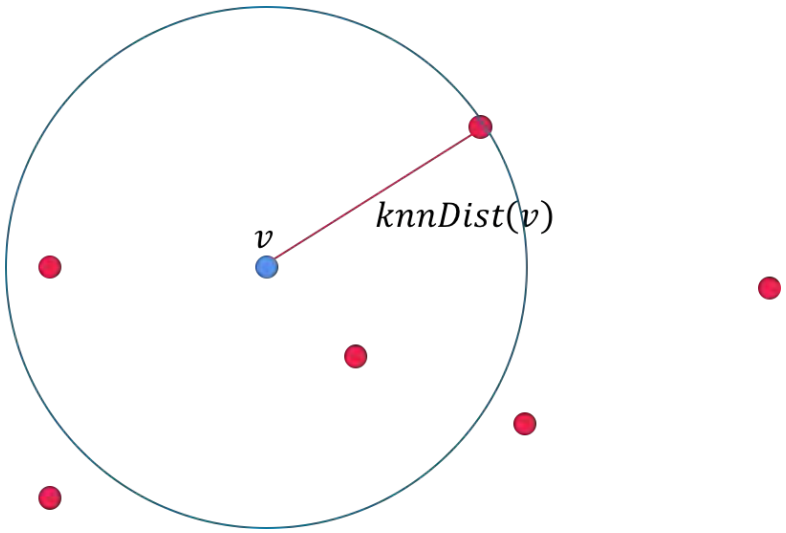


Figure 2.7: Example of calculating $knnDist$ of point v with $k = 4$.

2.3 Anomaly Detection on Time Series

In this section, the Seasonal Hybrid ESD (S-H-ESD) [15], the anomaly detection method introduced by Twitter, is reviewed.

2.3.1 Seasonal Hybrid ESD

The Seasonal Hybrid ESD is built upon generalized Extreme Studentized Deviate test (ESD). It utilizes a seasonal-trend decomposition procedure based on loess (STL) [8] to extract the time series $Y = (y_1, y_2, \ldots, y_n)$ into three compo-

nents: seasonality (s_i), trend (t_i), and remainder (y_i)

$$y_i = s_i + t_i + r_i.$$

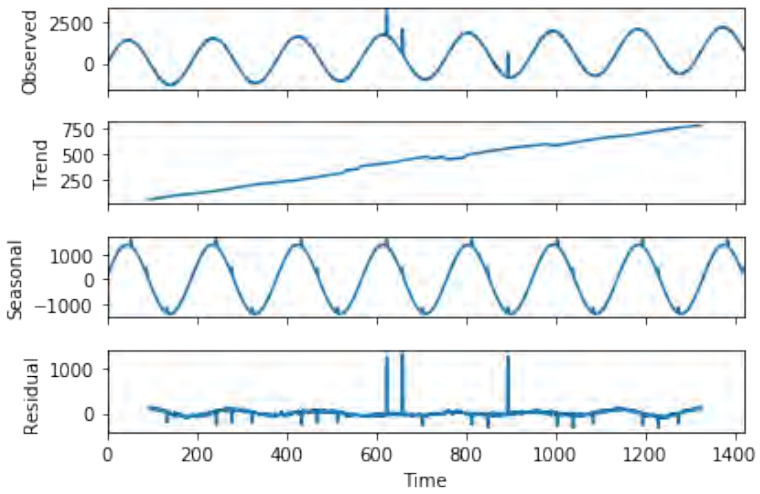


Figure 2.8: An example of STL performing on a synthetic time series data

To get the residual component, the time series is subtracted by its seasonal and trend component:

$$r_i = y_i - s_i - t_i.$$

The residual, then, is analyzed to find anomalous data in the time series by applying generalized extreme Studentized deviate test.

2.3.1.1 **Generalized Extreme Studentized Deviate**

A generalized Extreme Studentized Deviate (ESD) [20] is a method used to detect multiple anomalies in a univariate data set. This method requires a value k, which is the upper bound on the number of anomalies, to be specified as a parameter. It performs by calculating the test statistic C_i for the k most extreme

values in the dataset by

$$C = \frac{\max_j |x_j - \bar{x}|}{s} \tag{2.11}$$

where s is standard deviation of the dataset.

Initially, C_1 is set to C with a complete data set. Then, the value C_1 will be compared against critical value α_1 defined by

$$\alpha_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i-1+t^2_{p,n-i-1})(n-i-1)}}$$

where $t_{p,n-i-1}$ is the upper critical values of the t-distribution with $n-i-1$ degree of freedom and a significance level of p.

If $C_1 > \alpha_1$, the value x_j corresponding to the equation (2.11) will be masked as an anomaly and removed from consideration. Otherwise, it stops. Next, $C_2, C_3, \ldots, C_k$ will be calculated consecutively in the same manner as before with successively reduced data set.

CHAPTER III

ANOMALY DETECTION ON TIME SERIES FROM FURTHEST NEIGHBOR WINDOW SUBSERIES

In this chapter, a novel algorithm for detecting contextual anomalies on time series data is proposed. It is called Furthest Neighbor Window Subseries algorithm (FNWS). The motivation, methodology and time complexity analysis of the algorithm are presented in this chapter.

3.1 Definitions

This section discusses necessary definitions that will be used in the FNWS algorithm.

3.1.1 Representative Vector

For any given window subseries S_i, there are two main features of each window subseries to be considered as a representation. The first feature is the representation of the distribution. If a window subseries has a different distribution, it is an evidence that there is anomaly occur in that window subseries. The second feature shows the perspective of that window subseries according to the first instance so that each window subseries will be observed from the same perspective. This feature does not effect by a linear trend in time series.

To achieve that effect, this research uses the subtraction of each window

subseries by its first value which makes all windows start at 0. The first feature is trickier to achieve since there are a lot of options that can represent the distribution of each window subseries. This research chooses a representation as a vector of three values: the lower quartile, the median, and the upper quartile of that window subseries.

Definition 3.1 (Representative Vector)**.** Given a time series T of length m and window subseries S_i of length n is defined by $S_i = (t_i, t_{i+1}, \ldots, t_{i+n-1})$ where $i \in \{1, 2, \ldots, m-n+1\}$. A representative vector rv_i for S_i is a vector of the lower quartile (Q_1), the median (Q_2), and the upper quartile (Q_3) of S_i subtracting the first data point t_i of S_i.

$$rv_i = (Q_1 - t_i, Q_2 - t_i, Q_3 - t_i) \tag{3.1}$$

The representative vector has many benefits. First, it reduces the data in window subseries to just three values, hence, it can be efficiently computed and manipulated. Another benefit is that it can be efficiently extracted when a sliding window technique is applied. Calculating the representative of the first window subseries of a time series requires $O(n \log n)$ time complexity, because it has to be sorted first for computing the lower quartile, the median, and the upper quartile. But for the next window subseries, it only requires $O(n)$ time complexity because there is $n-1$ sorted value, and only inserts a new value to the appropriate location and recomputing those three values.

3.2 Furthest Neighbor Window Subseries Search Algorithm

The FNWS algorithm composes of three main steps: 1) extracting the characteristic vector, 2) calculating scores, and 3) identifying anomalous data points.

Initialization: Given the window subseries size n.

1. For each window subseries $S_i = (t_i, t_{i+1}, \ldots, t_{i+n-1})$ in a time series dataset, compute the lower quartile Q_1, the median Q_2, the upper quartile Q_3. Then generate the representative vector $rv_i = (Q_1 - t_i, Q_2 - t_i, Q_3 - t_i)$. This representative vector captures the distribution from the viewpoint of t_i.

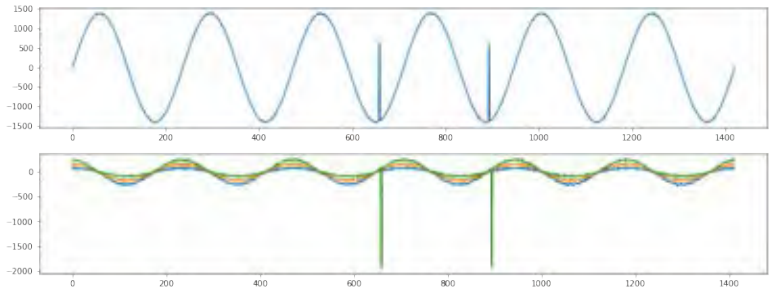


Figure 3.1: Example of a time series with anomalies going in the same direction (top) plot alongside with representative vector of each window subseries (bottom).

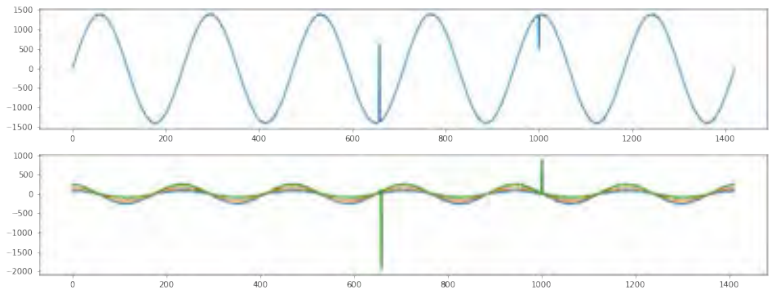
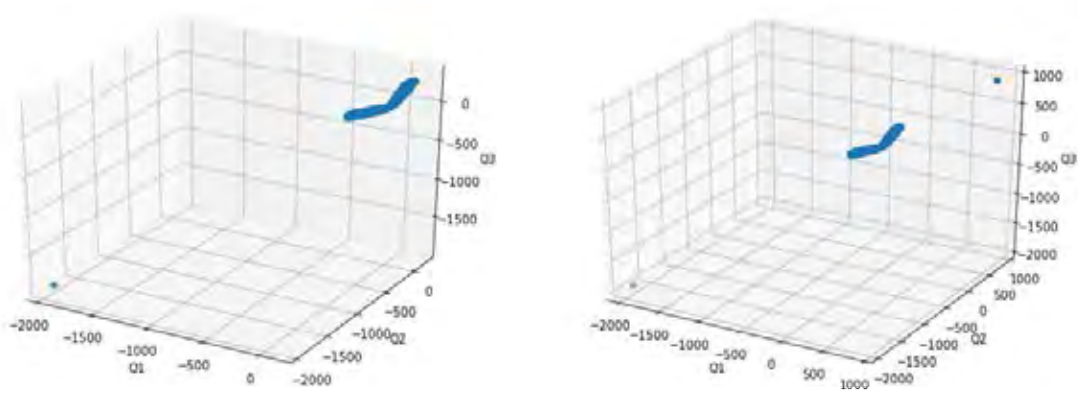


Figure 3.2: Example of a time series with anomalies go in opposite direction (top) plot alongside with representative vector of each window subseries (bottom).



(a) The first example (b) The second example

Figure 3.3: A 3-dimensional representation of the representative vectors of both examples.

2. Determine the k-nearest neighbor distance for each rv_i. Then compute $knnDist(rv_i)$. This value is a score for the window subseries.

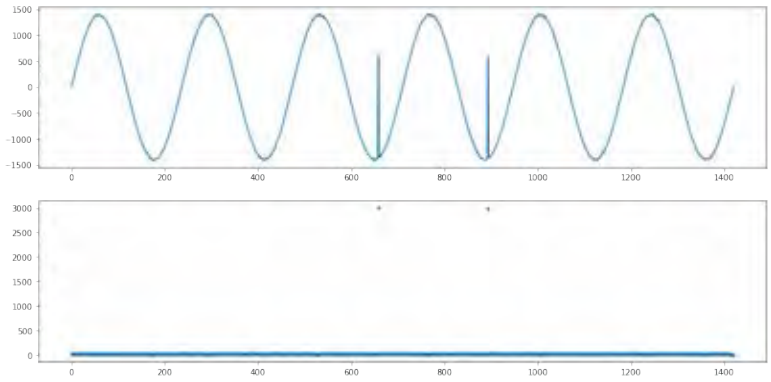


Figure 3.4: Example of a time series (top) with $knnDist$ of each window subseries (bottom).

Figure 3.4 exhibits a very high score for each anomalous data point while normal data points stay close to zero.

3. Calculate anomalous data points using interquartile range rule from the

adjusted boxplot for skew distributions.

$$anom\_cri = Q_3 + 1.5e^{3MC}IQR \tag{3.2}$$

where Q_3 is the upper quartile of scores and IQR is interquartile range of these scores and MC is *medcouple* which is defined by equation (2.7).

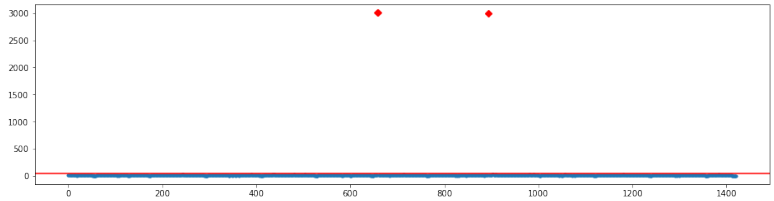


Figure 3.5: *knnDist* values of each window subseries with *anom_cri* line and points which are marked as anomalies.

Figure 3.5 shows data points above the anomaly criteria line which are identified as outliers.

3.3 Time Complexity Analysis

In this section, the time complexity of the FNWS algorithm is analyzed. The algorithm composes of three main steps:

1. extracting the characteristic vector,
2. calculating score, and
3. identifying anomalous data points.

3.3.1 Extracting the characteristic vector

In the first step, quartiles are extracted from the sorted window subseries of fixed size n having $m - n + 1$ elements. The algorithm performs $O(mn)$ running time.

3.3.2 Calculating score

In the process of calculating k-nearest neighbor distance, the nearest neighbor search in a K-D tree data structure [3] will be used which can compute the k-nearest neighbor distance much more efficient than the brute-force approach. K-D Tree algorithm constructs tree-based data structure for keeping the data to reduce the required number of distance calculations by efficiently encoding aggregate distance information for each data. The fundamental idea of this algorithm is that if point A is very far from point B, and point B is very close to point C, then A and C will be far away to each other, without having to explicitly calculate their distances. By using this algorithm the computational cost of the nearest search of each element can be reduced to $O(m \log(m))$ which is better than $O(m^2)$ in brute-force search.

3.3.3 Identifying anomalous data points

The last step of the algorithm is done by looping through all data points comparing it with the criteria in equation (3.2) which takes $O(m)$ time complexity. So overall time complexity is $O(m) + O(m \log m) + O(m) = O(m \log m)$.

CHAPTER IV

EXPERIMENTATION

In this chapter, the performance of the FNWS algorithm is tested and compared with S-H-ESD by using benchmark dataset from Yahoo. The FNWS algorithm is also tested on two different kinds of random data: white noise and random walk. All experiments and details are explained in this chapter.

4.1 Accuracy Measurement

In this section, three accuracy measurements, which is precision, recall, and F-measure, are used to evaluate the performance of the FNWS algorithm. These measurements can be derived easily from a confusion matrix.

4.1.1 Confusion Matrix

The confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model. Each column of the matrix represents the class of data instances from the model prediction while each row represents the actual class instances.

		Prediction	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 4.1: Confusion matrix

- True positive (TP): the number of instances correctly labeled as belonging to the positive class
- True negative (TN): the number of instances correctly labeled as belonging to the negative class
- False negative (FP): the number of instances incorrectly labeled as belonging to the positive class
- False negative (FN): the number of instances incorrectly labeled as belonging to the negative class

For the anomaly detection work, the positive class is the anomalous class and the negative class is the normal class.

4.1.2 **Precision**

The precision, also known as the positive predictive value, is the ratio between the number of the predicted instances as anomalies that are actual anomalies and the number of all data instances that are predicted as anomalies.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

4.1.3 **Recall**

The recall, also known as the true positive rate or sensitivity, is the ratio between the number of the predicted instances as anomalies that are actual anomalies and the number of all anomalous instances in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

4.1.4 F-measure

F-measure is the harmonic mean of the precision and recall.

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4.3}$$

4.2 Parameter Setting

There are only two parameters in the FNWS algorithm. The first parameter is the length of a window subseries n and another parameter is the number of the nearest neighbors k. The experiments vary n from 15 to 300. For the parameter k, the experiments set this value to the the length of the window subseries.

The S-H-ESD algorithm is also required two parameters. The first parameter is max_anoms; it is the maximum number of anomalies that S-H-ESD will detect as a percentage of the data. The max_anoms parameter is always set to 0.2. Another parameter is *period* that is used to define the number of observations in a single period, and is used during seasonal decomposition. This parameter is varied from 15 to 300.

4.3 FNWS on Synthetic Dataset

In this section, the FNWS algorithm is tested on benchmark datasets from a collection of Yahoo! Webscope datasets [23]. This collection consists of 100 synthetic time series data with varying trends, noises, and seasonality. All time series datasets are of length 1421 with the anomaly tag labels. The accuracy of the FNWS algorithm is measured by three metrics: precision, recall, and F-measure.

n, k	Precision	Recall	F-measure
5	0.8950 ± 0.1720	0.9858 ± 0.0682	0.9252 ± 0.1369
10	0.9852 ± 0.0949	0.9975 ± 0.0249	0.9876 ± 0.0755
15	0.9925 ± 0.0746	0.9942 ± 0.0412	0.9906 ± 0.0641
20	0.9848 ± 0.1075	0.9800 ± 0.0891	0.9748 ± 0.1045
25	0.9728 ± 0.1441	0.9656 ± 0.1356	0.9614 ± 0.1424

Table 4.2: Average ± standard deviation of precision, recall, and F-measure of the FNWS algorithm on 100 synthetic datasets with varying parameter from 5 to 25

4.4 **Result of S-H-ESD Algorithm**

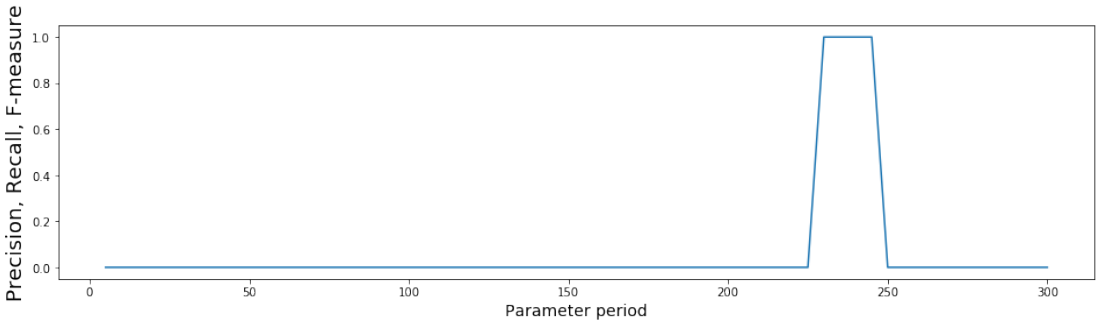


Figure 4.1: Precision, Recall and F-measure (solid line) of the S-H-ESD

While the results of the FNWS algorithm are quite stable and reach its best setting around $n = k = 10$ and 15 with precision, recall, and F-measure equal to 0.9925, 0.99416667 and 0.99057143, respectively, the S-H-ESD fails on other parameter settings and get only achieve its best performance when the *period* equal to $230, 235, 240$, and 245 with precision, recall, and F-measure scores are all 1 but it fails with other parameter settings. This means that the S-H-ESD is very sensitive to its parameters.

4.5 Test on Random Data

In this section, the FNWS algorithm is tested on two different datasets. The first one is white noise. The second is a random walk. Each dataset is generated from a uniform distribution and Gaussian distribution.

4.5.1 White Noise

White Noise process is a random process having equal intensity at different frequencies. A white noise can be seen as a sequence of random variables that have zero mean and finite variance and are statistically uncorrelated. Formally, $X(t)$ is a white noise process if

1. $E[X(t)] = 0$
2. $E[X^2(t)] = s^2$ for some $s \in \mathbb{R}$
3. $E[X(t_1)X(t_2)] = 0$ for $t_1 \neq t_2$

Examples of the simplest representatives of the white noise are independent and identically distributed (i.i.d.) random variables. In this work, white noises generated from Gaussian and uniform distribution will be used to test a behavior of the FNWS algorithm.

4.5.1.1 Uniform Distribution

This experiment tests FNWS algorithm on 1,000 white noises generated from a uniform distribution with minimum and maximum are equal to 0 and 1 respectively.

On average, the FNWS algorithm detects 17.24 points as anomalies with standard deviation equal to 3.13.

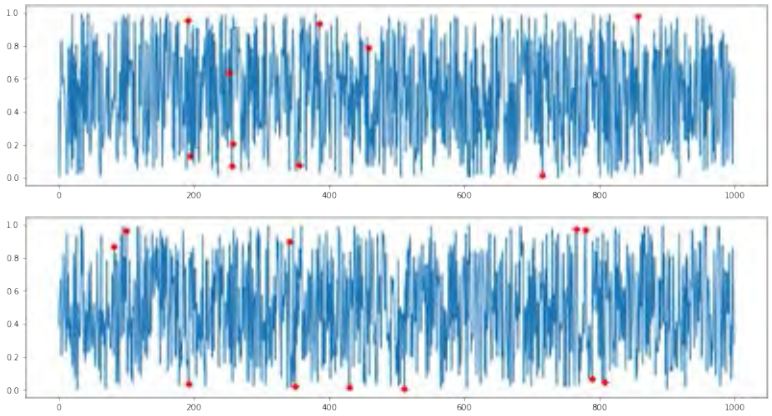


Figure 4.2: Examples of the FNWS algorithm performs on white noises generated from uniform distribution.

4.5.1.2 **Gaussian Distribution**

This experiment tests FNWS algorithm on 1,000 white noises generated from a Gaussian distribution with mean equal to 0 and standard deviation equal to 1.

On average, the FNWS algorithm detects 16.57 points as anomalies with standard deviation equal to 3.74.

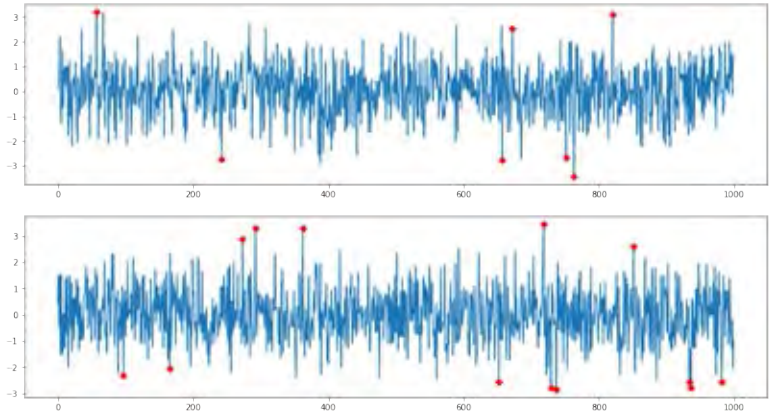


Figure 4.3: Examples of the FNWS algorithm performs on white noises generated from Gaussian distribution.

4.5.2 **Random Walk**

A random walk can be described as a path that consists of a sequence of random discrete steps. It can be interpreted as a sequence of the cumulative sum of random variables. [14]

Definition 4.1 (Random Walk)**.** Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of independent and identically distributed random variables. Let $S_0 = 0$ and for each $k \in \mathbb{N}$, S_k is defined to be

$$S_k = \sum_{i=1}^{k} X_i.$$

The sequence $(S_n)_{n\in\mathbb{N}}$ is called a *random walk*.

This work is interested in two types of univariate random walk that each step size and direction (X_i) are generated according to a Gaussian distribution and a uniform distribution.

4.5.2.1 **Uniform Distribution**

A thousand random walk is synthesized to test the FNWS algorithm. Each step of random walk is generated from a uniform distribution with maximum and minimum equal to -0.5 and 0.5 respectively.

On average, the FNWS algorithm detects 19.51 points as anomalies with standard deviation equal to 3.70.

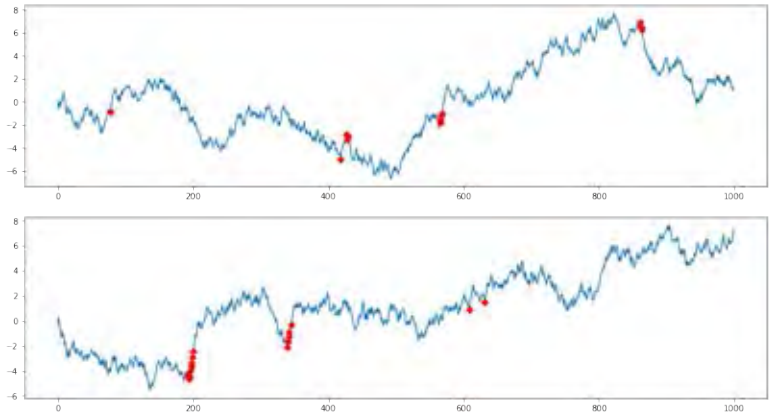


Figure 4.4: Examples of the FNWS algorithm performs on random walk generated from uniform distribution.

4.5.2.2 **Gaussian Distribution**

A thousand random walk is synthesized to test the FNWS algorithm. Each step of random walk is generated from a Gaussian distribution with mean equal to 0 and standard deviation equal to 1.

On average, the FNWS algorithm detects 19.97 points as anomalies with standard deviation equal to 3.95.

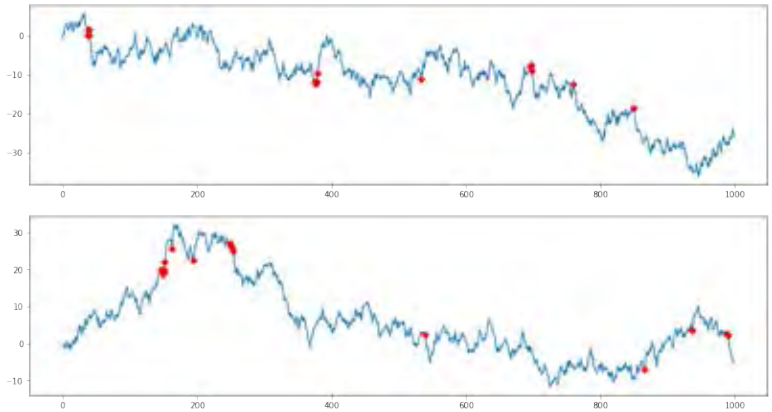


Figure 4.5: Examples of the FNWS algorithm performs on random walk generated from Gaussian distribution.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research begins with a review of the concept of anomaly detection in data mining in order to grasp and accumulate basic necessary information and various approaches for the solution of this problem. In particular, the literatures about processing time series data and the behavior and categories of anomalies in this type of data are examined. In this thesis, the new method, Furthest Neighbor Window Subseries (FNWS) algorithm, is proposed and demonstrated for detecting anomalies on time series of type one and of type two which are point anomaly and contextual anomaly.

The method is tested against the benchmark time series datasets provided by Yahoo! Webscope datasets and is compared the results with Seasonal Hybrid ESD (S-H-ESD), the method proposed by Twitter for detecting anomalies. The performance is evaluated with various metrics, namely, precision, recall and F-measure.

From the experiments, the S-H-ESD performs very well with the appropriate parameter setting but it failed in other situations, that is, when the parameter *periods* do not match the season of that time series while the FNWS algorithm shows the accuracy within [0.8-1] on all parameter settings.

5.2 Future work

In a future study, this method can be improved by investigating the two parameters being used in the algorithm. For the algorithm to work, it requires a setting of a window subseries size n which depends on a dataset. In addition, the number of the nearest neighborhoods, k, is needed to generate the appropriate score. In the future, these two parameters could be automated based on a given time series dataset.

References

[1] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185.

[2] David F Andrews and Agnes M Herzberg. *Data: a collection of problems from many fields for the student and research worker*. Springer Science & Business Media, 2012.

[3] Jon Louis Bentley. "Multidimensional Binary Search Trees Used for Associative Searching". In: *Commun. ACM* 18.9 (Sept. 1975), pp. 509–517. ISSN: 0001-0782. DOI: 10.1145/361002.361007. URL: http://doi.acm.org/10.1145/361002.361007.

[4] Markus M. Breunig et al. "LOF: Identifying Density-based Local Outliers". In: *SIGMOD Rec.* 29.2 (May 2000), pp. 93–104. ISSN: 0163-5808. DOI: 10.1145/335191.335388. URL: http://doi.acm.org/10.1145/335191.335388.

[5] G Brys, M Hubert, and A Struyf. "A Robust Measure of Skewness". In: *Journal of Computational and Graphical Statistics* 13.4 (2004), pp. 996–1017. DOI: 10.1198/106186004X12632. eprint: http://dx.doi.org/10.1198/106186004X12632. URL: http://dx.doi.org/10.1198/106186004X12632.

[6] N. Buthong, A. Luangsodsai, and K. Sinapiromsaran. "Outlier detection score based on ordered distance difference". In: *2013 International Computer Science and Engineering Conference (ICSEC)*. Sept. 2013, pp. 157–162. DOI: 10.1109/ICSEC.2013.6694771.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3 (July 2009), 15:1–15:58. ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: http://doi.acm.org/10.1145/1541880.1541882.

[8] Robert B Cleveland, William S Cleveland, and Irma Terpenning. "STL: A seasonal-trend decomposition procedure based on loess". In: *Journal of Official Statistics* 6.1 (1990), p. 3.

[9] Dipankar Dasgupta and Stephanie Forrest. "Novelty Detection in Time Series Data using Ideas from Immunology". In: *In Proceedings of The International Conference on Intelligent Systems*. 1995.

[10] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010.

[11] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "Advances in Knowledge Discovery and Data Mining". In: ed. by Usama M. Fayyad et al. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. Chap. From Data Mining to Knowledge Discovery: An Overview, pp. 1–34. ISBN: 0-262-56097-6. URL: http://dl.acm.org/citation.cfm?id=257938.257942.

[12] Ary L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet". In: *Circulation* 101.23 (2000), e215–e220. ISSN: 0009-7322. DOI: 10.1161/01.CIR.101.23.e215. eprint: http://circ.ahajournals.org/content/101/23/e215.full.pdf. URL: http://circ.ahajournals.org/content/101/23/e215.

[13] Markus Goldstein and Andreas Dengel. "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm". In: *KI-2012: Poster and Demo Track* (2012), pp. 59–63.

[14] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

[15] Jordan Hochenbaum, Owen S. Vallis, and Arun Kejariwal. "Automatic Anomaly Detection in the Cloud Via Statistical Learning". In: *CoRR* abs/1704.07706 (2017). URL: http://arxiv.org/abs/1704.07706.

[16] M. Hubert and E. Vandervieren. "An adjusted boxplot for skewed distributions". In: *Computational Statistics & Data Analysis* 52.12 (2008), pp. 5186–5201. ISSN: 0167-9473. DOI: http://doi.org/10.1016/j.csda.2007.11.008. URL: http://www.sciencedirect.com/science/article/pii/S0167947307004434.

[17] Eamonn Keogh, Jessica Lin, and Ada Fu. "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence". In: *Proceedings of the Fifth IEEE International Conference on Data Mining*. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 226–233. ISBN: 0-7695-2278-5. DOI: 10.1109/ICDM.2005.79. URL: http://dx.doi.org/10.1109/ICDM.2005.79.

[18] L. E. Peterson. "K-nearest neighbor". In: *Scholarpedia* 4.2 (2009). revision #136646, p. 1883. DOI: 10.4249/scholarpedia.1883.

[19] David M W Powers. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63.

[20] Bernard Rosner. "Percentage points for a generalized ESD many-outlier procedure". In: *Technometrics* 25.2 (1983), pp. 165–172.

[21] Owen Vallis, Jordan Hochenbaum, and Arun Kejariwal. "A Novel Technique for Long-term Anomaly Detection in the Cloud". In: *Proceedings of the 6th USENIX Conference on Hot Topics in Cloud Computing*. Hot-

Cloud'14. Philadelphia, PA: USENIX Association, 2014, pp. 15–15. URL: http://dl.acm.org/citation.cfm?id=2696535.2696550.

[22] William Wu-Shyong Wei. *Time series analysis*. Addison-Wesley publ Reading, 1994.

[23] Yahoo. *A Labeled Anomaly Detection Dataset, version 1.0*. URL: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s%7B%5C&%7Ddid=70 (visited on 02/08/2016).

Biography

Name	Mr. Senee Kitimoon
Date of Birth	22 April 1992
Place of Birth	Chiang Mai, Thailand
Education	B.S. (Mathematics) (First Class Honours), Chiang Mai University, 2013
Scolarship	Development and Promotion for Science and Technology talent (DPST)