ปัจจัยการตรวจหาข้อมูลที่แตกต่างจากข้อมูลอื่นที่ไร้พารามิเตอร์โดยใช้ค่าถ่วงน้ำหนักต่ำสุดของคู่ที่ติดกัน

นางสาววรัญญา เกียงเอีย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

PARAMETER-FREE OUTLIER DETECTION FACTOR USING WEIGHTED MINIMUM CONSECUTIVE PAIR

Miss Warunya Kiangia

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	PARAMETER-FREE OUTLIER DETECTION FACTOR USING WEIGHTED MINIMUM CONSECUTIVE PAIR
By	Miss Warunya Kiangia
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Arthorn Luangsodsai, Ph.D.
Thesis Co-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

(Asssociate Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

. . . . Chairman

(Assistant Professor Phantipa Thipwiwatpotjana, Ph.D.)

. . . . Thesis Advisor

(Arthorn Luangsodsai, Ph.D.)

. . . . Thesis Co-advisor

(Assistant Professor Krung Sinapiromsaran, Ph.D.)

. . . . Examiner

(Kitiporn Plaimas, Ph.D.)

. . . . External Examiner

(Assistant Professor Chumphol Bunkhumpornpat, Ph.D.)

วรัญญา เกียงเอีย : ปัจจัยการตรวจหาข้อมูลที่แตกต่างจากข้อมูลอื่นที่ไร้พารามิเตอร์โดยใช้ค่าถ่วงน้ำหนักต่ำสุดของคู่ที่ติดกัน. (PARAMETER-FREE OUTLIER DETECTION FACTOR USING WEIGHTED MINIMUM CONSECUTIVE PAIR) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ดร. อาธร เหลืองสดใส, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร. กรุง สินอภิรมย์สราญ 53 หน้า.

แนวคิดการตรวจหาข้อมูลที่แตกต่างจากข้อมูลอื่นเป็นหนึ่งในหัวข้อสำคัญที่สนใจศึกษาในการทำเหมืองข้อมูล งานวิจัยต่างๆ เกี่ยวกับการระบุข้อมูลที่แตกต่างจากข้อมูลอื่นมุ่งเน้นการสร้างขั้นตอนวิธีการคำนวณคะแนนของข้อมูลที่แตกต่างจากข้อมูลอื่น ซึ่งสามารถนำไปวัดความแตกต่างจากข้อมูลอื่นของตัวอย่างในเซตข้อมูล ออร์เดอร์ดีสเตนดิฟเฟอร์เรนซ์เอาท์ไลน์เออร์แฟคเตอร์หรือโอโอเอฟ เป็นขั้นตอนวิธีที่ไร้พารามิเตอร์สำหรับการตรวจหาข้อมูลที่แตกต่างจากข้อมูลอื่น ตีพิมพ์ในปี คศ. 2013 วิทยานิพนธ์นี้นำเสนอขั้นตอนวิธีไร้พารามิเตอร์ที่เรียกว่า เวททิดมินนิมัมคอนเซคคิวทีฟแพร์ออฟดิเอ็กตรีมโพเอาท์ไลเออร์แฟคเตอร์หรือ ดับเบิ้ลยูโอเอฟ การให้คะแนนข้อมูลที่แตกต่างจากข้อมูลอื่นใหม่ของตัวอย่างถูกสร้างขึ้นตามข้อมูลที่ไกลที่สุดสองตัวอย่างโดยพิจารณา ภาพฉายรัศมีของตัวอย่างนี้และตัวอย่างคู่ที่ต่อเนื่องตามลำดับ จำนวนน้อยที่สุดในแต่ละด้านของตัวอย่างจะถูกถ่วงน้ำหนัก และใช้ในการสร้างดับเบิ้ลยูโอเอฟ ขั้นตอนวิธีดับเบิ้ลยูโอเอฟมีความซับซ้อนของเวลาเป็นบิ๊กโอเอ็นกำลังสอง เพื่อเปรียบเทียบประสิทธิภาพและเวลา ขั้นตอนวิธีดับเบิ้ลยูโอเอฟถูกใช้กับเซตข้อมูลจำลองที่สร้างขึ้นและเซตข้อมูลยูซีไอสามเซต

ภาควิชา	คณิตศาสตร์และ	ลายมือชื่อนิสิต ....
	วิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
สาขาวิชา	คณิตศาสตร์ประยุกต์	ลายมือชื่อ อ.ที่ปรึกษาร่วม ....
	และวิทยาการคณนา	
ปีการศึกษา	2559	

5772136623 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : OUTLIER DETECTION / EXTREME POLES / ANOMALY DETECTION

WARUNYA KIANGIA : PARAMETER-FREE OUTLIER DETECTION FACTOR USING WEIGHTED MINIMUM CONSECUTIVE PAIR. ADVISOR : ARTHORN LUANGSODSAI, Ph.D., COADVISOR : ASSISTANT PROFESSOR KRUNG SINAPIROMSARAN, Ph.D., 53 pp.

Outlier concept is one of the most significant topics in data mining. Many researches in outlier detections address an algorithm to generate the outlier scores which can be used to measure the outlierness of an instance in a dataset. Ordered distance difference outlier factor (OOF) is the parameter-free outlier detection algorithm which was published in 2013. This thesis proposes a new parameter-free outlier detection algorithm called a weighted minimum consecutive pair of the extreme pole outlier factor (WOF). The new outlier score of an instance is generated along the extreme poles by considering the radial projection of this instance and its consecutive pair. The minimum on each side of the instance will be weighted and used to create the WOF. The WOF algorithm has the $O(n^2)$ time complexity. To compare the effectiveness and time, WOF algorithm was applied with generated synthetic datasets and three UCI datasets.

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

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Arthorn Luangsodsai and my co-advisor Assistant Professor Dr. Krung Sinapiromsaran for the continuous support in this thesis. When problems arose, they always gave many suggestions until I finished this work for the master program. I could not complete this thesis without their helpful suggestions.

Next, I would like to thank my thesis committee, Assistant Professor Dr. Phantipa Thipwiwatpotjana, Dr. Kitiporn Plaimas and Assistant Professor Dr. Chumphol Bunkhumpornpat for their comments and suggestions.

Moreover, I would like to thank the program of Applied Mathematics and Computational Science (AMCS) in the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University for funding scholarship and technical supports.

Finally, I am thankful to my family and my friends in AMCS laboratory especially Panote Songwattanasiri, Suebkul Kanchanasuk, Sanee Kitimoon and everybody for all their supports throughout the period of this thesis.

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Outlier detection is an important topic in data mining as an outlier indicates abnormal point that varies difference from most remaining points in a dataset. Hawkins's defined the notion of an outlier as follows [1] : "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". Outlier detection has been used in many real world applications such as fraudulent detections, intrusion detections, machine failure detections, fault detection in safety critical systems, military surveillances for enemy activity and insurance or health care [2].

There have been many research works on novel outlier detections which use a variety of techniques and theories [3, 4, 5, 6]. In 1998, Knorr and Ng [3] proposed a distance-based outlier detection. If the neighbors within the radius q of an instance contains more than n instances then it is called an inlier, otherwise it is called an outlier. The advantage of this concept is its simplicity to detect the outliers by Hawkins' definition. However, it cannot detect outliers when a dataset has various densities. In 2000, Breuning et al. [4] proposed a density-based algorithm called local outlier factor (LOF). The concept of the LOF based on a local density where locality is given by $k-$nearest neighbors. The outlier score for an instance is calculated by comparing the local density of an instance to the local density of its neighbors. The LOF is a prototype for many forthcoming published papers. For example, in 2002, Tang et al. [7] introduced a technique called connectivity-based outlier factor (COF) that remedies the weakness of LOF. They assigned each instance an outlier score by the ratio of the average chaining distances from the instance to the chaining of its neighbors.

In 2012, Goldstein and Dengle [8] introduced a new algorithm that worked in $O(n)$

linear time called the histogram-based outlier factor (HBOS). They assigned an outlier score for each instance by a fixed bin-width histogram and the height of a bin represents a density estimate. In 2013, Buthong et al. [5] proposed the different approach to LOF called the ordered distance difference outlier factor. It relies on the distance of an instance to the extreme object with respect to itself using the minimum of the ordered distance differences from this instance along the pole of its extreme object. Then the outlier score is assigned as the ratio of this minimum with the number of instances excluding itself. The process of computing OOF uses the distance matrix having $O(n^2 \log n)$ time complexity.

This thesis introduces a weighted minimum consecutive pair outlier factor (WOF) which calculates an outlier score along two extreme poles. All instances are projected in the core vector built from them. WOF is the average of two distances from each extreme pole where the distances are generated between the instance and the first neighbors on each side of the core vector which are weighted by the number of instances in the same side. The WOF algorithm is the parameter-free algorithm and has $O(n^2)$ time complexity.

1.1 Research Objectives

The goal of this research is to obtain a new parameter-free outlier detection algorithm called the weighted minimum consecutive pair of the extreme pole outlier factor or WOF. The WOF algorithm is implemented and its performance and time complexity are compared with OOF on three synthetic datasets and three real world datasets.

1.2 Thesis Overview

Chapter II describes the background knowledge such as the metric measure, the extreme pole, the meaning of an outlier and the algorithm for calculating an outlier score. Next, the ordered distance difference outlier factor is explained. In Chapter III, the weighted minimum consecutive pair of the extreme pole outlier factor is presented with its definitions and algorithm. The performance comparison with OOF are presented in Chapter IV. Finally, Chapter V gives the conclusion of this work.

CHAPTER II

PRELIMINARIES

This chapter describes the background knowledge and the main concept for this thesis based on distance matrix and extreme poles. This chapter is divided into four parts. The first part gives the definition of a metric space and the distance function. The second part gives the definition of the extreme poles. The third part defines the outlier detection types and discusses Ordered distance difference Outlier Factor or OOF. Finally, the fourth part shows the definition and the algorithm of OOF with an example to illustrate this outlier score.

2.1 Metric

This section shows the definition of a metric space and a distance matrix. A distance function is a function that defines a distance between a pair of elements in a dataset. A set together with a distance function is called a metric space.

Definition 2.1. (Metric space) [9]

Let B be an arbitrary set. A metric space is an ordered pair (B, d) where a function $d : B \times B \longrightarrow \mathbb{R}^+ \bigcup \{0\}$ is a metric on B such that for any $u, v, w \in B$,

1) $d(u,v) \geq 0$ (Positiveness)

2) $d(u,v) = 0$ if and only if $u = v$ (Identity)

3) $d(u,v) = d(v,u)$ (Symmetry)

4) $d(u,w) \geq d(u,v) + d(v,w)$ (Triangle inequality).

This definition characterizes the distance between two instances and the definition of the distance matrix. Let B be a set and u, v be two data points in B. The function d is called the distance function. $d(u,v)$ means the distance between the instance u and v.

Definition 2.2. (The Minkowski distance between two instances) [10]

Given dataset $D \subseteq \mathbb{R}^m$ of n instances, $p \in D$ is an instance with m attributes, and $p^{(i)}$ for $i \in \{1, 2, 3, ..., n\}$ is the i^{th} instance. The Minkowski distance of order k between two instances p and q where $p = (p_1, p_2, p_3, ..., p_m)$ and $q = (q_1, q_2, q_3, ..., q_m)$ is defined as

$$d_k(p, q) = \sqrt[k]{\sum_{j=1}^{m} |p_j - q_j|^k},$$

We called the Manhattan distance if we set $k = 1$, it is written as

$$d_1(p, q) = \sum_{j=1}^{m} |p_j - q_j|.$$

This thesis uses the Euclidean distance by setting $k = 2$,

$$d_2(p, q) = \sqrt{\sum_{j=1}^{m} (p_j - q_j)^2}.$$

To simplify the notation, d will be used to represent the Euclidean distance.

Definition 2.3. (The distance matrix) [11]

The matrix of distances between instances from dataset D is defined by

$$M = [d_{i,j}]_{n \times n},$$

such that $d_{i,j} = d(p^{(i)}, p^{(j)})$, where $p^{(i)}, p^{(j)} \in D$ and $i, j \in \{1, 2, 3, ..., n\}$.

Definition 2.3 gives the calculation of the distance between instances in a dataset which is represented by the following matrix.

$$M = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & \cdots & d_{1,n} \\ d_{2,1} & 0 & d_{2,3} & \cdots & d_{2,n} \\ d_{3,1} & d_{3,2} & 0 & \cdots & d_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & d_{n,3} & \cdots & 0 \end{bmatrix}.$$

Next, dataset A is randomly generated to help explaining the definition.

Example 2.1. Dataset A has two instances $p \in (1, 2)$ and $q \in (7, 10)$.

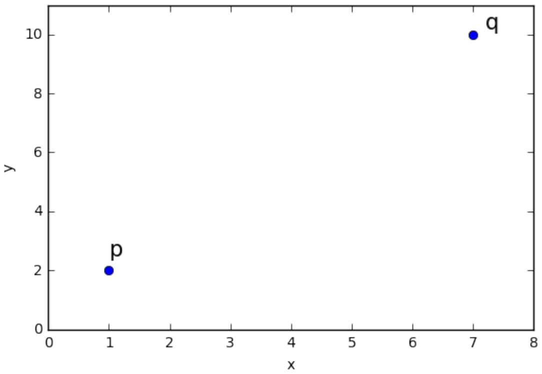


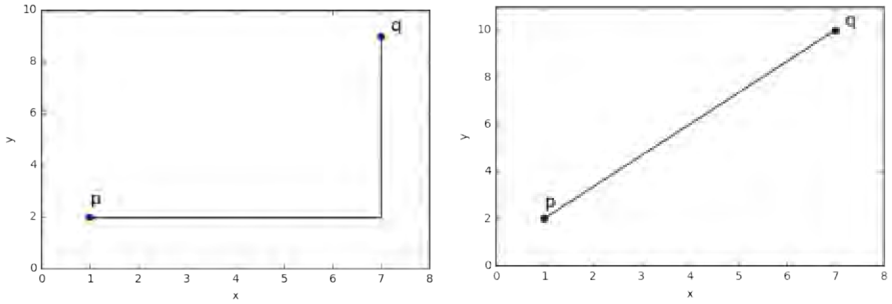
Figure 2.1: The generated points in dataset A

The Manhattan distance between two instances p and q is

$$d_1(p, q) = \sum_{j=1}^{2} |p_j - q_j| = |1 - 7| + |2 - 10| = 6 + 8 = 14.$$

The Euclidean distance between instances p and q is

$$d_2(p, q) = \sqrt{\sum_{j=1}^{2} (p_j - q_j)^2} = \sqrt{(1 - 7)^2 + (2 - 10)^2} = \sqrt{6^2 + 8^2} = \sqrt{100} = 10.$$



(a) Manhattan distance

(b) Euclidean distance

Figure 2.2: Manhattan distance vs Euclidean distance

Figure 2.2 shows the result of the different metrics between the Manhattan distance and the Euclidean distance.

Example 2.2**.** Dataset B (Figure 2.3) has six randomly generated instances in $\mathbb{R}^2$.

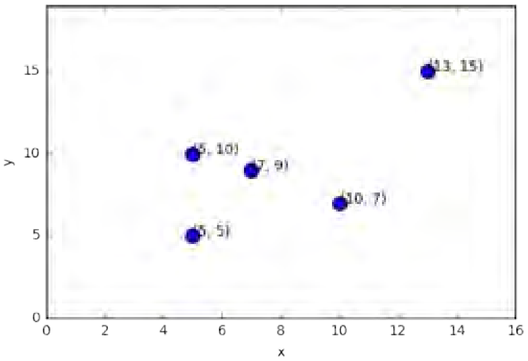


Figure 2.3**:** The randomly generated points in dataset B

The Euclidean distance between any two instances in dataset B can be represented as matrix M as follows :

$$M = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0.0 & 5.0 & 4.47 & 5.39 & 12.81 & 12.17 \\ 5.0 & 0.0 & 2.24 & 5.83 & 9.43 & 12.37 \\ 4.47 & 2.24 & 0.0 & 3.61 & 8.49 & 10.20 \\ 5.39 & 5.83 & 3.61 & 0.0 & 8.54 & 7.0 \\ 12.81 & 9.44 & 8.49 & 8.54 & 0.0 & 8.94 \\ 12.17 & 12.40 & 10.20 & 7.0 & 8.94 & 0.0 \end{bmatrix}.$$

2.2 Extreme Poles

In this section, the concept of extreme poles and a core vector are explained. The extreme poles originally were used for the classification and the clustering [12, 13]. The extreme poles are the pair of instances that make the largest separation. The definition of an extreme pole is explained next.

Definition 2.4. (The extreme poles)

Given $e_1 \in \{1, 2, 3, ..., n\}$ and $e_2 \in \{1, 2, 3, ..., n\}$ such that

$$d(p^{(e_1)}, p^{(e_2)}) = \max_{i,j} d(p^{(i)}, p^{(j)})$$

$p^{(e_1)}$ and $p^{(e_2)}$ are called the extreme poles. Note that for a finite data set, there exists at least two extreme poles.

Figure 2.4. shows two extreme poles by Definition 2.4. Moreover, the extreme poles appear at the rim of the intended region of a dataset.

Definition 2.5. (The core vector)

The core vector is a vector that starts from one extreme pole to another extreme pole.

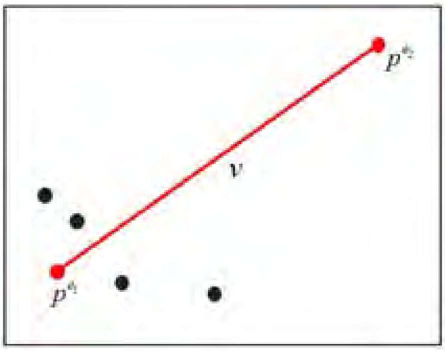


Figure 2.4: p^{e_1} and p^{e_2} are extreme poles and v is the vector core

Figure 2.4, p^{e_1} and p^{e_2} are the poles. It is the farthest pair among all instances in the dataset and the vector v is defined as the core vector of the dataset.

2.3 Outlier

This section introduces the outlier techniques, types of outiers and outlier detection. In a dataset, an instance that is placed far from most instances or does not conform to a notion of normal patterns is called outlier, see Hawkins [1].

2.3.1 Outlier Detection Approaches

Three popularly uses of outlier detection techniques are the statistics-based, the distance-based and the density-based approaches.

2.3.1.1 The Statistics-Based Approach

The statistic approach assumes a probability or a distribution model for a dataset. Then, an outlier is identified by the model using the discordancy test. Many techniques are only applicable in one dimension. If the dimension increases, it becomes difficult and inaccurate to identify an outlier in the multidimensional space, see [14].

2.3.1.2 The Density-Based Approach

The density-based approach estimates the density distribution based on neighborhood of each instance. If the neighbors of an instance has similar density, then the instance is identified as the normal instance, see [4].

2.3.1.3 The Distance-Based Approach

The distance-based approach calculates an outlier as an instance which has a large distance from other instances, see [5, 7, 8].

2.3.2 Outlier Detection

Next, the outlier detection is described that depends on the availability of labels in a training dataset. 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, an outlier detection algorithm can operate in one of the following three modes.

2.3.2.1 Supervised Outlier Detection

This type of algorithm applies to a training dataset which all instances have been labeled as normal or outlier. Typically, a predictive model is used to distinguish an instance into the normal and outlier class. Any unseen data instance is compared against the model to determine which class it belongs to. There are two major issues that arise in supervised outlier detection. First, the anomalous instances are few comparing with the normal instances in the training data. Second, predicting accurate labels for both normal and outlier classes is usually challenging.

2.3.2.2 Semi-Supervised Outlier Detection

In this group of a semi-supervised algorithm, a training dataset assumes to have only the normal class label. Then the model is built based on these instances. Instances that do not comply with the model are reported as the outlier. Since they do not require labels for the outlier class, they are more widely applicable than supervised outlier detection.

2.3.2.3 Unsupervised Outlier Detection

This situation does not require a training dataset, and thus are most widely applicable. The techniques in this category make the implicit assumption that normal instances are far more frequent than outliers. If this assumption is not true then such techniques suffer from high false alarm rate.

2.4 Ordered Distance Difference Outlier Factor

This section explains OOF which shares the common characteristic and is used to compare performances with our algorithm. In 2013, Buthong.N et al. [5] proposed a parameter-free distance-based method, called ordered distance difference outlier (OOF).

Definition 2.6. (The Minimum Distance of p)

Given the dataset D with n instances and $p \in D$

$$mindist(p) := min\{d(p,q)|q \in D\backslash\{p\}\}$$

Then, $mindist(p)$ be the minimum distance of between p and other instances.

Definition 2.7. (The Difference Distance between q and o w.r.t p)

Let $p, q, o \in D$, the difference distance between the instances q and o with respect to the instance p when p is fixed as a common instance is defined by the difference between $d(p,q)$ and $d(p,o)$. It is written as,

$$\triangle d_p(q,o) := |d(p,q) - d(p,o)|.$$

Figure 2.5 shows the difference distance between two instances with respect to instance p. it means the difference between the distance p, q or $d(p,q)$ and the distance p, o or $d(p,o)$. The value of this equation is always positive.

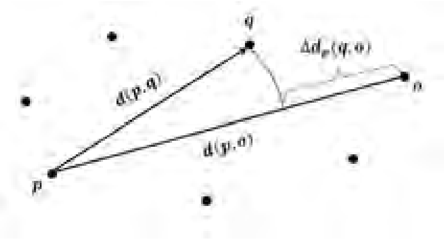


Figure 2.5**:** The difference distance between two instances with respect to instance p

Definition 2.8. (The Distance Matrix)

The distance matrix of the data set D is defined as:

$$DistMtx(D) := (d_{i,j})_{n \times n},$$

such that $d_{i,j} = d(p^{(i)}, p^{(j)})$, where $p^{(i)}, p^{(j)} \in D$ and $i, j \in \{1, 2, 3, ..., n\}$.

Definition 2.9. (The Ordered Distance Matrix)

The ordered distance matrix of the dataset D is defined by

$$OrderedMtx(D) := (\vec{O})_{n \times 1},$$

where $i \in \{1, 2, ..., n\}$ and $(\vec{O})_1$ is an ordered distance of row i^{th} of the distance matrix. This is defined by $(\vec{O})_1 = (d_{i,j_1^{(i)}} \; d_{i,j_2^{(i)}} \; \ldots \; d_{i,j_k^{(i)}} \; \ldots \; d_{i,j_n^{(i)}})$, where $k, j_k^{(i)} \in \{1, 2, 3, ..., n\}$ with $d_{i,j_1^{(i)}} \leq d_{i,j_2^{(i)}} \leq ... \leq d_{i,j_n^{(i)}}$.

The distance matrix calculates the distance from an instance to all other instances which is represented by the following matrix:

$$DistMtx(D) = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots & d_{1,n} \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots & d_{2,n} \\ d_{3,1} & d_{3,2} & d_{3,3} & \cdots & d_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & d_{n,3} & \cdots & d_{n,n} \end{bmatrix}.$$

Each row in the $OrderedMtx$ from the definition 2.9 is ascending sorted using the nearest neighbor idea. Let $d_{i,j_k^{(i)}}$ be the distance between the instance $p^{(i)}$ and k^{th} nearest neighbors of $p^{(i)}$ and the distance of $d_{i,i}$ is zero. Consequently, the ordered distance matrix is generated as

$$OrderedMtx(D) = \begin{bmatrix} 0 & d_{1,j_1^{(1)}} & d_{1,j_2^{(1)}} & d_{1,j_3^{(1)}} & \cdots & d_{1,j_n^{(1)}} \\ 0 & d_{2,j_1^{(2)}} & d_{2,j_2^{(2)}} & d_{2,j_3^{(2)}} & \cdots & d_{2,j_n^{(2)}} \\ 0 & d_{3,j_1^{(3)}} & d_{3,j_2^{(3)}} & d_{3,j_3^{(3)}} & \cdots & d_{3,j_n^{(3)}} \\ 0 & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & d_{n,j_1^{(n)}} & d_{n,j_2^{(n)}} & d_{i,j_3^{(n)}} & \cdots & d_{i,j_n^{(n)}} \end{bmatrix}.$$

Definition 2.10. (The Difference of the Ordered Distance Matrix)

The difference of the ordered distance matrix of dataset D is defined by

$$\Delta OrderedMtx(D) := (\Delta\vec{O}_i)_{n\times 1},$$

where $i \in \{1, 2, 3, ..., n\}$ and $(\Delta\vec{O}_i)$ is the difference of the ordered distance of row i^{th} of the distance matrix.

Definition 2.11**.** (The Difference of the Ordered Distance Outlier Factor)

The difference of the ordered distance outlier factor (OOF) of instance p is defined by

$$OOF := \frac{\sum_{i=1}^{n} min\{\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}), mindist(p)\}}{n-1}.$$

An instance p in the ordered distance difference outlier factor is calculated without parameter. It relies on the distance from p to the extreme pole using the minimum of the ordered distance difference from this instance along the pole with its extreme pole. Then the outlier score is assigned as the ratio of this minimum with the number of instances excluding itself. If its score is high, then it has a high probability to be outlier. Next example shows how to calculate OOF of all instances in the dataset.

Example 2.3**.** Given dataset C that contains 6 instances having one majority group and a single outlier. $C = [(1,5),(2,4),(2,5),(2,6),(3,4),(9,13)]$.

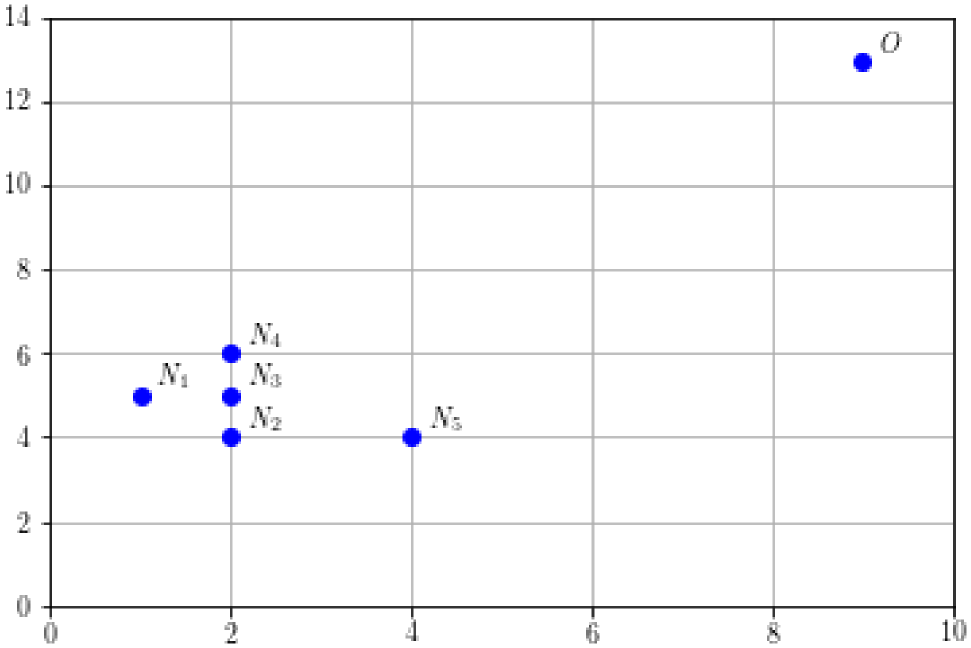


Figure 2.6**:** The dataset C containing 6 instances with a single outlier

Dataset C is randomly generated having one majority group with the single outlier (O). Only the calculation of three instances N_1, N_3, O are demonstrated in details.

The mathematical notation for the distance matrix is

$$DistMtx(C) = \begin{bmatrix} d(N_1,N_1) & d(N_1,N_2) & d(N_1,N_3) & d(N_1,N_4) & d(N_1,N_5) & d(N_1,O) \\ d(N_2,N_1) & d(N_2,N_2) & d(N_2,N_3) & d(N_2,N_4) & d(N_2,N_5) & d(N_2,O) \\ d(N_3,N_1) & d(N_3,N_2) & d(N_3,N_3) & d(N_3,N_4) & d(N_3,N_5) & d(N_3,O) \\ d(N_4,N_1) & d(N_4,N_2) & d(N_4,N_3) & d(N_4,N_4) & d(N_4,N_5) & d(N_4,O) \\ d(N_5,N_1) & d(N_5,N_2) & d(N_5,N_3) & d(N_5,N_4) & d(N_5,N_5) & d(N_5,O) \\ d(O,N_1) & d(O,N_2) & d(O,N_3) & d(O,N_4) & d(O,N_5) & d(O,O) \end{bmatrix},$$

$$= \begin{bmatrix} 0.0 & 1.4142 & 1.0 & 1.4142 & 2.2361 & 11.3137 \\ 1.4142 & 0.0 & 1.0 & 2.0 & 1.0 & 11.4018 \\ 1.0 & 1.0 & 0.0 & 1.0 & 1.4142 & 10.6301 \\ 1.4142 & 2.0 & 1.0 & 0.0 & 2.2361 & 9.8995 \\ 2.2361 & 1.0 & 1.4142 & 2.2361 & 0.0 & 10.8167 \\ 11.3137 & 11.4018 & 10.6301 & 9.899 & 10.8167 & 0.0 \end{bmatrix}.$$

The distance matrix is sorted by the distance in each row which generates the ordered distance matrix

$$\vec{O}(C) = \begin{bmatrix} d_1(N_1,N_1) & d_1(N_1,N_3) & d_1(N_1,N_2) & d_1(N_1,N_4) & d_1(N_1,N_5) & d_1(N_1,O) \\ d_2(N_2,N_2) & d_2(N_2,N_3) & d_2(N_2,N_5) & d_2(N_2,N_1) & d_2(N_2,N_4) & d_2(N_2,O) \\ d_3(N_3,N_3) & d_3(N_3,N_1) & d_3(N_3,N_2) & d_3(N_3,N_4) & d_3(N_3,N_5) & d_3(N_3,O) \\ d_4(N_4,N_4) & d_4(N_4,N_3) & d_4(N_4,N_1) & d_4(N_4,N_2) & d_4(N_4,N_5) & d_4(N_4,O) \\ d_5(N_5,N_5) & d_5(N_5,N_2) & d_5(N_5,N_3) & d_6(N_5,N_1) & d_5(N_5,N_4) & d_5(N_5,O) \\ d_6(N_6,N_6) & d_6(N_6,N_4) & d_6(N_6,N_3) & d_6(N_6,N_5) & d_6(N_6,N_1) & d_6(N_1,O) \end{bmatrix},$$

$$= \begin{bmatrix} 0.0 & 1.0 & 1.4142 & 1.4142 & 2.2361 & 11.3137 \\ 0.0 & 1.0 & 1.0 & 1.4142 & 2.0 & 11.4018 \\ 0.0 & 1.0 & 1.0 & 1.0 & 1.4142 & 10.6301 \\ 0.0 & 1.0 & 1.4142 & 2.0 & 2.2367 & 9.8995 \\ 0.0 & 1.0 & 1.4142 & 2.2361 & 2.2361 & 10.8167 \\ 0.0 & 9.8995 & 10.6301 & 10.8167 & 11.3137 & 11.4018 \end{bmatrix}.$$

The result of the computation is

$$\Delta\vec{O}(C) = \begin{bmatrix} 0 & \Delta d_1(N_3,N_1) & \Delta d_1(N_2,N_3) & \Delta d_1(N_4,N_2) & \Delta d_1(N_5,N_4) & \Delta d_1(O,N_5) \\ 0 & \Delta d_2(N_3,N_2) & \Delta d_2(N_5,N_3) & \Delta d_2(N_1,N_5) & \Delta d_2(N_4,N_1) & \Delta d_2(O,N_4) \\ 0 & \Delta d_3(N_1,N_3) & \Delta d_3(N_2,N_1) & \Delta d_3(N_4,N_2) & \Delta d_3(N_5,N_4) & \Delta d_3(O,N_5) \\ 0 & \Delta d_4(N_3,N_4) & \Delta d_4(N_1,N_3) & \Delta d_4(N_2,N_1) & \Delta d_4(N_5,N_2) & \Delta d_4(O,N_5) \\ 0 & \Delta d_5(N_2,N_5) & \Delta d_5(N_3,N_2) & \Delta d_5(N_1,N_3) & \Delta d_5(N_4,N_1) & \Delta d_5(O,N_4) \\ 0 & \Delta d_6(N_4,N_6) & \Delta d_6(N_3,N_4) & \Delta d_6(N_5,N_3) & \Delta d_6(N_1,N_5) & \Delta d_6(N_2,N_1) \end{bmatrix},$$

$$= \begin{bmatrix} 0 & 1.0 & 0.4142 & 0.0 & 0.8219 & 9.0776 \\ 0 & 1.0 & 0.0 & 0.4142 & 0.5857 & 9.4017 \\ 0 & 1.0 & 0.0 & 0.0 & 0.4142 & 9.2159 \\ 0 & 1.0 & 0.4142 & 0.5857 & 0.2360 & 7.6634 \\ 0 & 1.0 & 0.4142 & 0.8218 & 0.0 & 8.5805 \\ 0 & 9.8994 & 0.7306 & 0.1865 & 0.4970 & 0.0880 \end{bmatrix}.$$

Next, the distance from a common instance to another instances on the ordered distance difference matrix for calculating OOF is considered.

Case N_1 :

$$\Delta\vec{O}(C) = \begin{bmatrix} \underline{0} & \Delta d_1(N_3,N_1) & \Delta d_1(N_2,N_3) & \Delta d_1(N_4,N_2) & \Delta d_1(N_5,N_4) & \Delta d_1(O,N_5) \\ 0 & \Delta d_2(N_3,N_2) & \Delta d_2(N_5,N_3) & \underline{\Delta d_2(N_1,N_5)} & \Delta d_2(N_4,N_1) & \Delta d_2(O,N_4) \\ 0 & \underline{\Delta d_3(N_1,N_3)} & \Delta d_3(N_2,N_1) & \Delta d_3(N_4,N_2) & \Delta d_3(N_5,N_4) & \Delta d_3(O,N_5) \\ 0 & \Delta d_4(N_3,N_4) & \underline{\Delta d_4(N_1,N_3)} & \Delta d_4(N_2,N_1) & \Delta d_4(N_5,N_2) & \Delta d_4(O,N_5) \\ 0 & \Delta d_5(N_2,N_5) & \Delta d_5(N_3,N_2) & \underline{\Delta d_5(N_1,N_3)} & \Delta d_5(N_4,N_1) & \Delta d_5(O,N_4) \\ 0 & \Delta d_6(N_4,N_6) & \Delta d_6(N_3,N_4) & \Delta d_6(N_5,N_3) & \underline{\Delta d_6(N_1,N_5)} & \Delta d_6(N_2,N_1) \end{bmatrix}.$$

Then the ordered distance difference outlier factor or OOF is calculated:

$$\begin{aligned} OOF(N_1) = [&min\{0, mindist(N_1)\} + min\{\Delta d_2(N_1,N_5), mindist(N_1)\} \\ &+ min\{\Delta d_3(N_1,N_3), mindist(N_1)\} + min\{\Delta d_4(N_1,N_3), mindist(N_1)\} \\ &+ min\{\Delta d_5(N_1,N_3), mindist(N_1)\} + min\{\Delta d_6(N_1,N_5), mindist(N_1)\}]/6. \end{aligned}$$

Note $mindist(p)$ is the minimum distance of p

(i.e. $mindist(p) := min\{d(p,q)|q \in D\backslash\{p\}\}$). Then, $mindist(N_1) = 1$. Hence,

$$\begin{aligned} OOF(N_1) &= \frac{0 + 0.4142 + 1.0 + 0.4142 + 0.8218 + 0.4970}{6} \\ &= \frac{3.1472}{6} \\ &= 0.5245. \end{aligned}$$

Case N_3 **:**

$$\Delta\vec{O}(C) = \begin{bmatrix} 0 & \underline{\Delta d_1(N_3,N_1)} & \Delta d_1(N_2,N_3) & \Delta d_1(N_4,N_2) & \Delta d_1(N_5,N_4) & \Delta d_1(O,N_5) \\ 0 & \underline{\Delta d_2(N_3,N_2)} & \Delta d_2(N_5,N_3) & \Delta d_2(N_1,N_5) & \Delta d_2(N_4,N_1) & \Delta d_2(O,N_4) \\ \underline{0} & \Delta d_3(N_1,N_3) & \Delta d_3(N_2,N_1) & \Delta d_3(N_4,N_2) & \Delta d_3(N_5,N_4) & \Delta d_3(O,N_5) \\ 0 & \underline{\Delta d_4(N_3,N_4)} & \Delta d_4(N_1,N_3) & \Delta d_4(N_2,N_1) & \Delta d_4(N_5,N_2) & \Delta d_4(O,N_5) \\ 0 & \Delta d_5(N_2,N_5) & \underline{\Delta d_5(N_3,N_2)} & \Delta d_5(N_1,N_3) & \Delta d_5(N_4,N_1) & \Delta d_5(O,N_4) \\ 0 & \Delta d_6(N_4,N_6) & \underline{\Delta d_6(N_3,N_4)} & \Delta d_6(N_5,N_3) & \Delta d_6(N_1,N_5) & \Delta d_6(N_2,N_1) \end{bmatrix}.$$

Then the ordered distance difference outlier factor or OOF is calculated:

$$\begin{aligned} OOF(N_3) = [&min\{\Delta d_1(N_3,N_1), mindist(N_3)\} + min\{\Delta d_2(N_3,N_2), mindist(N_3)\} \\ &+ min\{0, mindist(N_3)\} + min\{\Delta d_4(N_3,N_4), mindist(N_3)\} \\ &+ min\{\Delta d_5(N_3,N_2), mindist(N_3)\} + min\{\Delta d_6(N_3,N_4), mindist(N_3)\}]/6. \end{aligned}$$

From, $mindist(p)$ is the minimum distance of the instance p

(i.e. $mindist(p) := min\{d(p,q)|q \in D\backslash\{p\}\}$). Then, $mindist(N_3) = 1$. Hence,

$$\begin{aligned} OOF(N_3) &= \frac{1.0 + 1.0 + 0 + 1.0 + 0.4142 + 0.7306}{6} \\ &= \frac{4.1448}{6} \\ &= 0.6908. \end{aligned}$$

Case O:

$$\Delta\vec{O}(C) = \begin{bmatrix} 0 & \Delta d_1(N_3,N_1) & \Delta d_1(N_2,N_3) & \Delta d_1(N_4,N_2) & \Delta d_1(N_5,N_4) & \underline{\Delta d_1(O,N_5)} \\ 0 & \Delta d_2(N_3,N_2) & \Delta d_2(N_5,N_3) & \Delta d_2(N_1,N_5) & \Delta d_2(N_4,N_1) & \underline{\Delta d_2(O,N_4)} \\ 0 & \Delta d_3(N_1,N_3) & \Delta d_3(N_2,N_1) & \Delta d_3(N_4,N_2) & \Delta d_3(N_5,N_4) & \underline{\Delta d_3(O,N_5)} \\ 0 & \Delta d_4(N_3,N_4) & \Delta d_4(N_1,N_3) & \Delta d_4(N_2,N_1) & \Delta d_4(N_5,N_2) & \underline{\Delta d_4(O,N_5)} \\ 0 & \Delta d_5(N_2,N_5) & \Delta d_5(N_3,N_2) & \Delta d_5(N_1,N_3) & \Delta d_5(N_4,N_1) & \underline{\Delta d_5(O,N_4)} \\ \underline{0} & \Delta d_6(N_4,N_6) & \Delta d_6(N_3,N_4) & \Delta d_6(N_5,N_3) & \Delta d_6(N_1,N_5) & \Delta d_6(N_2,N_1) \end{bmatrix}.$$

Then the ordered distance difference outlier factor or OOF is calculated:

$$\begin{aligned} OOF(O) = [&min\{\Delta d_1(O,N_5), mindist(O)\} + min\{\Delta d_2(O,N_4), mindist(O)\} \\ &+ min\{\Delta d_3(O,N_5), mindis(O)\} + min\{\Delta d_4(O,N_5), mindist(O)\} \\ &+ min\{\Delta d_5(O,N_4), mindist(O)\} + min\{0, mindist(O)\}]/6. \end{aligned}$$

From, $mindist(p)$ is the minimum distance of the instance p (i.e. $mindist(p) := min\{d(p,q)|q \in D\backslash\{p\}\}$). Then, $mindist(O) = 9.899$. Hence,

$$\begin{aligned} OOF(O) &= \frac{9.0776 + 9.4017 + 9.2159 + 7.6634 + 8.5805 + 0}{6} \\ &= \frac{43.9391}{6} \\ &= 7.3232. \end{aligned}$$

This is the calculation of OOF scores for three instances, N_1, N_3, O. For other instances, the OOF scores of $N_2 = 0.4176, N_5 = 0.3317, N_4 = 0.3171$ are computed using python language on a web-based cloud computing named CoCalc.com[15]. Since the OOF score of O is 7.3232 which is significantly higher than other instances, that mean it should be considered as an outlier.

CHAPTER III

WEIGHTED MINIMUM CONSECUTIVE PAIR OF THE EXTREME POLE OUTLIER FACTOR

In this chapter, the definition of WOF, time complexity of WOF, and the WOF algorithm are described. Let D be a dataset having n instances with m attributes, $d(p, q)$ be the Euclidean distance between two instances p and q, $M(D)$ be the distance matrix of dataset D, and the extreme pole be the maximum distance between two instances in $M(D)$.

3.1 Definitions of Weighted Minimum Consecutive Pair of the Extreme Pole Outlier Factor

The weighted minimum consecutive pair of the extreme pole outlier factor (WOF) does not require any parameter and have $O(n^2)$ time complexity for assigning to all instances in the dataset. From the extreme pole of the dataset, WOF of each instance p is the weighted summation of the distance between p and its adjacent instances on the core vector. Since the core vector is generated from two extreme poles, WOF is set as the average computation from each extreme pole.

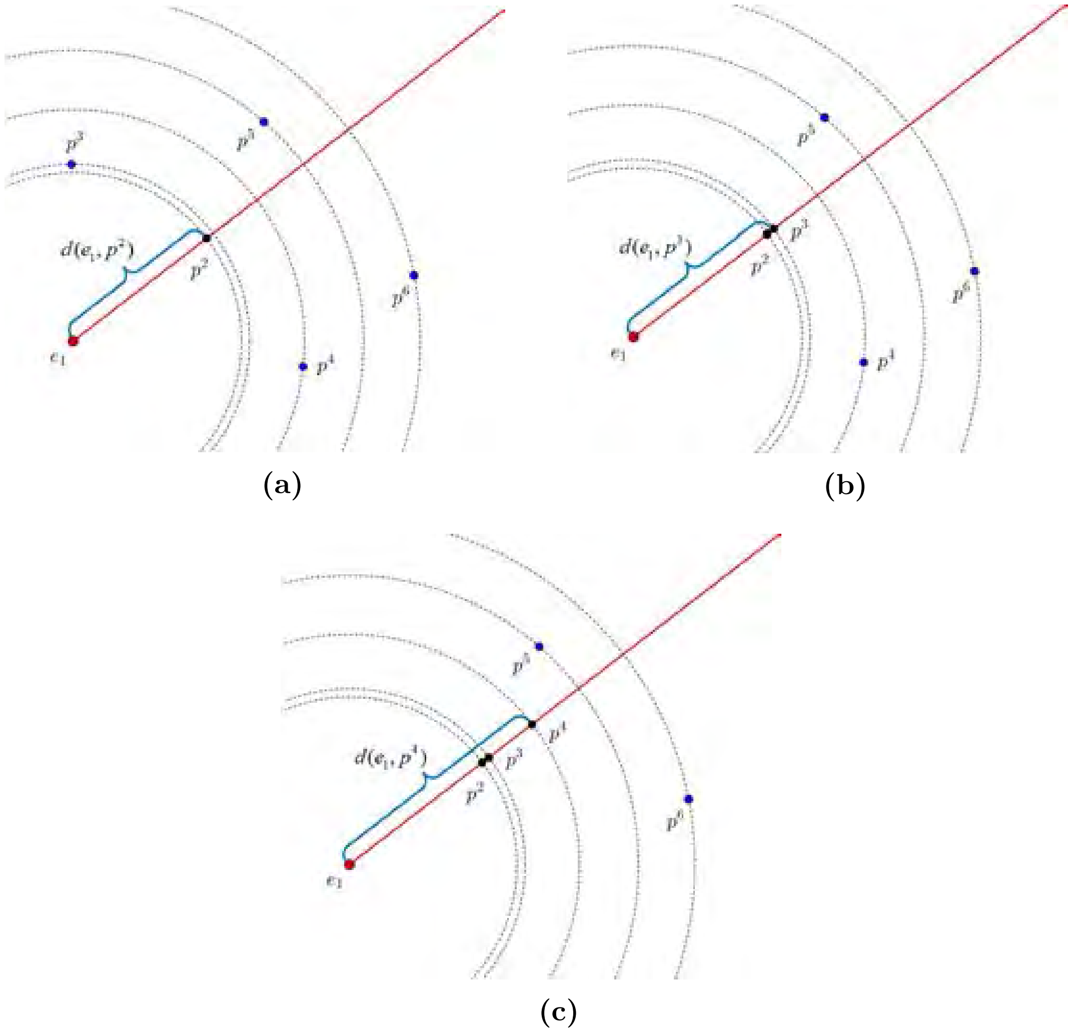
Using the property of the extreme poles and the core vector that is the distribution of all instances in dataset make acute angle between two vectors, the core vector and the vector generated from the extreme pole to an instance. Considering the radial projection of all instances on the core vector using one of the extreme poles as the center (see Figure 3.1). Figure 3.1(a), Figure 3.1(b) and Figure 3.1(c) show the distance from the radial projection from center $p^{(e_1)}$ to p^2, p^3, p^4, respectively which are ordered with respect to the center $p^{(e_1)}$.

Definition 3.1. (The radial projected order list on the core vector from the extreme pole)

The radial projected order list on the core vector from an extreme pole, e, of dataset D is defined by

$$OrdList(D,e) := (d_{(e,k)})_{1\times n},$$

where $e \in \{e_1, e_2\}$ is the index of the extreme pole and $k \in \{1,2,3,...,n\}$ with $0 = d_{(e,1)} \leq d_{(e,2)} \leq d_{(e,3)} \leq ... \leq d_{(e,k)} \leq ... \leq d_{(e,n)}$.



(a)

(b)

(c)

Figure 3.1: The radial projected order list on the core vector from the extreme pole

The distance between each instance is identified as the normal instance or outlier

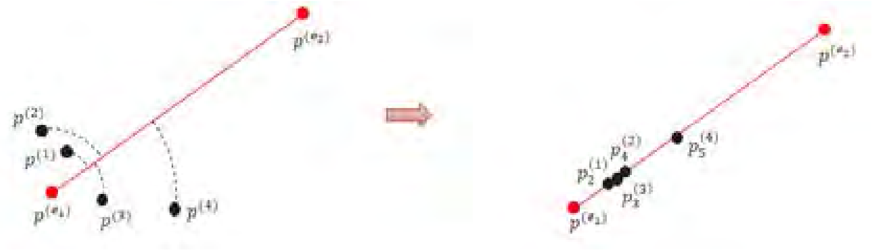


Figure 3.2: The radial projection of all other instances with respect to $p^{(e_1)}$

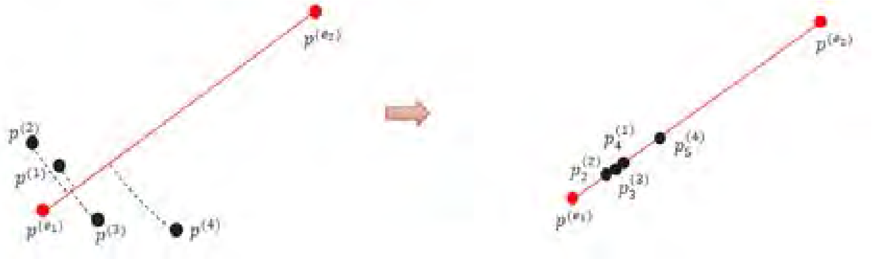


Figure 3.3: The radial projection of all other instances with respect to $p^{(e_2)}$

where the instance that is far away from other instances, tends to be outlier and the instance that is close to other instances, tends to be normal. Hence, the distance between the instance to the nearest instance represents the distance between instance to all instances toward the center or away from the center. Considering instance $p^{(k)}$ on the core vector that using the radial projection by center $p^{(e_1)}$ see Figure 3.4, $(d_{(e_1,p^{(k)})}-d_{(e_1,p^{(k-1)})})$ to represent the distance between the instance $p^{(k)}$ to the other instance within the radius $p^{(k)}$ and $(d_{(e_1,p^{(k+1)})}-d_{(e_1,p^{(k)})})$ to represent the distance between $p^{(k)}$ and other instances outside the radial projection. Then, the outlier factor is the weighted sum of all distances. Similarly, instance $p^{(k)}$ on the core vector that using the radial projection by center $p^{(e_2)}$ see Figure 3.4, $(d_{(e_2,p^{(k)})}-d_{(e_2,p^{(k-1)})})$ to represent the distance between the instance $p^{(k)}$ to the other instance within the radius $p^{(k)}$ and $(d_{(e_2,p^{(k+1)})}-d_{(e_2,p^{(k)})})$ to represent the distance between $p^{(k)}$ and other instances outside the radial projection. Then, the outlier factor is the weighted sum of all distances which is defined in Definition 3.2.

Definition 3.2. (The ordered outlier factor)

The ordered outlier factor of instance p computing by extreme pole e is defined by

$$OF_e(p^{(k)}) := \frac{(d_{(e,p^{(k)})}-d_{(e,p^{(k-1)})})(k-1)}{(n-1)} + \frac{(d_{(e,p^{(k+1)})}-d_{(e,p^{(k)})})(n-k)}{(n-1)},$$

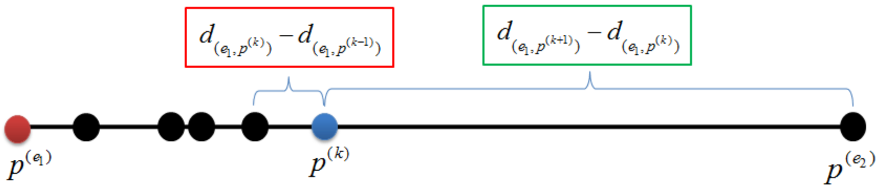

$d_{(e_1,p^{(k)})} - d_{(e_1,p^{(k-1)})}$
$d_{(e_1,p^{(k+1)})} - d_{(e_1,p^{(k)})}$
$p^{(e_1)}$
$p^{(k)}$
$p^{(e_2)}$

Figure 3.4**:** The radial projected order score on the core vector from $p^{(e_1)}$

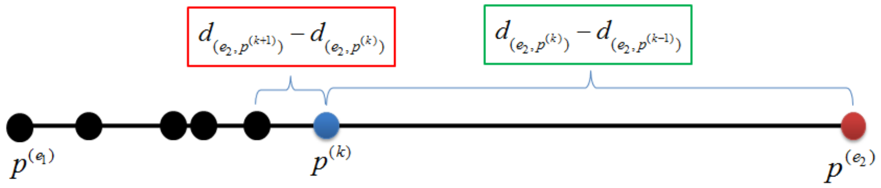
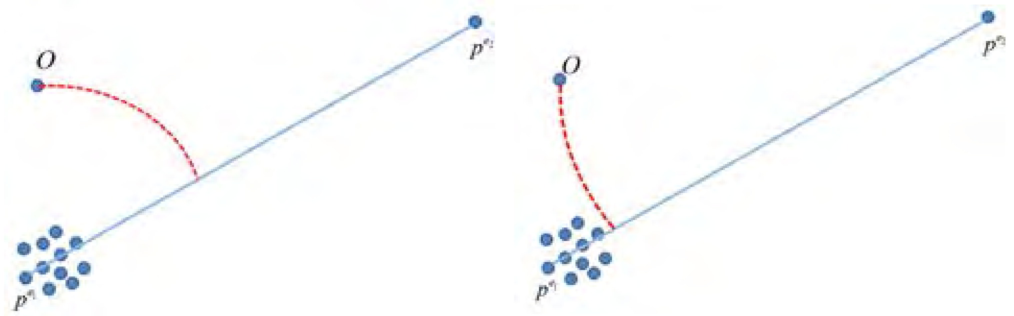

$d_{(e_2,p^{(k+1)})} - d_{(e_2,p^{(k)})}$
$d_{(e_2,p^{(k)})} - d_{(e_2,p^{(k-1)})}$
$p^{(e_1)}$
$p^{(k)}$
$p^{(e_2)}$

Figure 3.5**:** The radial projected order score on the core vector from $p^{(e_2)}$

where $k \in \{1, 2, 3, ..., n\}$.



(a) The radial projection the instance out of group with respect to $p^{(e_1)}$

(b) The radial projection the instance out of group with respect to $p^{(e_2)}$

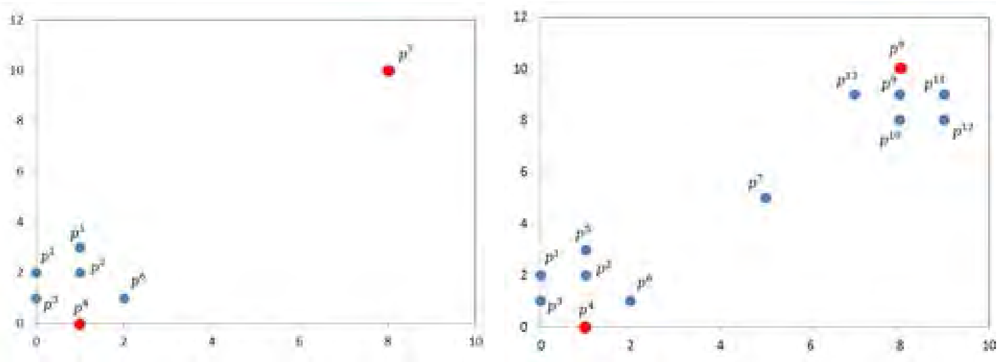
Figure 3.6**:** The radial projection the instance out of group with respect to extreme pole

That radial projected order list depends on the center either from the extreme poles e_1 or e_2. Figure 3.3 shows the radial projection of the dataset containing one group with two additional outliers. In Figure 3.6, the extreme poles are e_1 and e_2. Considering instance O, if this instance projected using the radial projection from the center e_2 to the core vector see Figure 3.6(a), it is close to the group of e_1 which is hard to identify as the outlier. If instance O is projected using the radial projection from the center e_1 to the core vector see Figure 3.6(b), it is far from the group of e_1 which can be detected as the outlier. Hence, the formula for WOF will compute from both e_1 and e_2 which is the average between $OF_{e_1}(p^{(k)})$ and $OF_{e_2}(p^{(k)})$.

Definition 3.3. (Weighted minimum consecutive pair of the extreme pole outlier factor)

The weighted minimum consecutive pair of the extreme pole outlier factor (WOF) is defined as

$$WOF(p) := \frac{OF_{e_1}(p^k) + OF_{e_2}(p^k)}{2}.$$



(a) Synthetic dataset A **(b)** Synthetic dataset B

Figure 3.7**:** Example of synthetic datasets

Example 3.1**.** Dataset A (Figure 3.7(a)) contains 7 instances having one majority group and a single outlier. $A = [(0,2),(1,2),(0,1),(1,0),(1,3),(2,1),(8,10)]$.

Dataset A is generated having one majority group with the single outlier (O) manually. The distance matrix of A is

$$DistMtx(A) = \begin{bmatrix} 0 & 1 & 1 & 2.236 & 1.414 & 2.236 & 11.313 \\ 1 & 0 & 1.414 & 2 & 1 & 1.414 & 10.630 \\ 1 & 1.414 & 0 & 1.414 & 2.236 & 2 & 12.041 \\ 2.236 & 2 & 1.414 & 0 & 3 & 1.414 & \boxed{12.206} \\ 1.414 & 1 & 2.236 & 3 & 0 & 2.236 & 9.899 \\ 2.236 & 1.414 & 2 & 1.414 & 2.236 & 0 & 10.816 \\ 11.313 & 10.630 & 12.041 & \boxed{12.206} & 9.899 & 10.816 & 0 \end{bmatrix}.$$

From the above distance matrix, two extreme poles are $p^{(4)}$ and $p^{(7)}$. Consider the instance $p^{(1)}$, the distance matrix is sorted by the row $p^{(4)}$ starting from $p^{(4)}$ to $p^{(7)}$ and sorted the row $p^{(7)}$ starting from $p^{(7)}$ to $p^{(4)}$.

$$\begin{aligned} OrdList(p^{(4)}) &= [p^{(4)} \quad p^{(3)} \quad p^{(6)} \quad p^{(2)} \quad p^{(1)} \quad p^{(5)} \quad p^{(7)}] \\ &= [0 \quad 1.414 \quad 1.414 \quad 2 \quad 2.236 \quad 3 \quad 12.206]. \\ OrdList(p^{(7)}) &= [p^{(7)} \quad p^{(5)} \quad p^{(2)} \quad p^{(6)} \quad p^{(1)} \quad p^{(3)} \quad p^{(4)}] \\ &= [0 \quad 9.899 \quad 10.630 \quad 10.816 \quad 11.313 \quad 12.041 \quad 12.206]. \end{aligned}$$

The calculations of $OF_{p^{(4)}}(p^{(1)})$ and $OF_{p^{(7)}}(p^{(1)})$ are

$$\begin{aligned} OF_{p^{(4)}}(p^{(1)}) &= \frac{(2.2360679774 - 2)(4) + (3 - 2.2360679774)(2)}{6} \\ &= 0.53934466. \\ OF_{p^{(7)}}(p^{(1)}) &= \frac{(11.313708 - 10.816653)(4) + (12.041594 - 11.313708)(2)}{6} \\ &= 0.69531282. \end{aligned}$$

$$\begin{aligned} \text{Hence, } WOF &= \frac{OF_{p^{(4)}}(p^{(1)}) + OF_{p^{(2)}}(p^{(1)})}{2} \\ &= \frac{0.53934466 + 0.69531282}{2} \\ &= 0.61732874. \end{aligned}$$

For other instances, WOFs of dataset A are shown in Table 3.1 computed using python language on COCALC.

instance	score
$p^{(1)}$	0.61732874
$p^{(2)}$	0.42462275
$p^{(3)}$	0.44863050
$p^{(4)}$	0.78958730
$p^{(5)}$	3.04301429
$p^{(6)}$	0.45638958
$p^{(7)}$	9.55302528

Table 3.1**:** WOFs of all instances in dataset A

Example 3.2**.** Given dataset B as in Figure 3.7(b) contains 10 instances having two group and a single outlier. $B = [(0, 2), (1, 2), (0, 1), (1, 0), (1, 3), (2, 1), (5, 5), (8, 10), (8, 9), (8, 8), (9, 9), (9, 8), (7, 9)]$.

WOFs of all instances in dataset B are shown in Table 3.2.

The examples of synthetic datasets A and B are displayed in Figure 3.7. From Table 3.1,

instance	score	instance	score
$p^{(1)}$	0.62391104	$p^{(8)}$	0.58248052
$p^{(2)}$	0.45399109	$p^{(9)}$	0.36543922
$p^{(3)}$	0.40628677	$p^{(10)}$	1.46566486
$p^{(4)}$	0.78958730	$p^{(11)}$	0.3415247
$p^{(5)}$	2.66284836	$p^{(12)}$	1.44854526
$p^{(6)}$	0.42126840	$p^{(13)}$	0.40980008
$p^{(7)}$	4.16904172		

Table 3.2: WOFs of all instances in dataset B

$p^{(7)}$ has the highest WOF in the dataset A which means that $p^{(7)}$ should be considered as the outlier. The instances with the lower WOF lies within the cluster such as $p^{(2)}, p^{(3)}, p^{(6)}$. From Table 3.2, $p^{(7)}$ has the highest WOF in the dataset B which means that $p^{(7)}$ should be considered as the outlier whereas instances $p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}, p^{(6)}, p^{(8)}$, $p^{(9)}, p^{(11)}, p^{(13)}$ are considered as inliers.

3.2 Time Complexity Analysis

For the time complexity analysis, the WOF algorithm is divided into two parts: 1) computing the distance matrix and 2) finding the extreme pole and computing the WOF. For the first part, the instances in the dataset have n instances then the distance matrix using $O(n^2)$ time complexity. In the second part, all row in the distance is found the extreme pole, then this part uses $O(n^2)$ time complexity. Then, the overall time complexity is $O(n^2) + O(n^2) = O(n^2)$.

3.3 Weighted minimum consecutive pair of the extreme pole outlier factor algorithm

Next, the WOF algorithm for computing the outlier score for all instances in the dataset is shown as follows :

```
INPUT:     Dataset $D$ with $n$ instances $m$ attributes
OUTPUT:    WOF for each instance
WOF Algorithm
  STEP 1:  Compute the distance between the instance $p^{(i)}$ and $p^{(j)}$ for every
           $i, j \in \{1, 2, 3, ..., n\}$ to build the distance matrix $(M)$
  STEP 2:  Find two extreme poles $p^{e_1}$ and $p^{e_2}$ in $D$ and construct the core vector
  STEP 3:  Generate the projected order list on the core vector from the extreme pole
  STEP 4:  Compute OF for each pole by Definition 3.2,
           $OF_e(p^{(k)}) := \frac{(d_{(e,p^{(k)})} - d_{(e,p^{(k-1)})})(k-1) + (d_{(e,p^{(k+1)})} - d_{(e,p^{(k)})})(n-k)}{(n-1)}$
           where $k \in \{1, 2, 3, ..., n\}$
  STEP 5:  Compute WOF of each instance by Definition 3.3, $WOF(p) := \frac{OF_{e_1} + OF_{e_2}}{2}$
```

CHAPTER IV

EXPERIMENTS AND RESULT

This chapter is divided into two sections. The first section describes the performance of the WOF algorithm comparing with the OOF algorithm and the second section covers the time complexity between the WOF algorithm and the OOF algorithm.

4.1 The Performance of the WOF Algorithm

This section explains the synthetic datasets and UCI datasets, the measurements for testing the performance and the experimental results.

4.1.1 Dataset

Three synthetic datasets in two-dimensional are generated and three real-world UCI datasets namely Wisconsin diagnostic breast cancer dataset, statlog (Landsat Satellite) dataset and glass identification dataset are used for testing the WOF algorithm. The details of all datasets that use in this thesis are shown in Table 4.1.

4.1.2 Measurements

In this part, the comparison results of the synthetic dataset are shown in the scatter plot and the comparison results of the real world dataset are shown using the area under the ROC curve[16, 17]. The ROC curve represents the relative trade-offs between the true positive rates (TPR) on the y-axis and false positive rate (FPR) on the x-axis. Since TPR is equivalent to sensitivity and FPR is equal to $1 -$ specificity. TPR is used for measuring the function of positive examples that are correctly labeled (4.1) and FPR is used to measure the function of negative examples that are misinterpreted as positive (4.2). Those measures can be derived from the confusion matrix in Table 4.2. The rows

Dataset	Attribute	Instance	Class	Class Name
Synthetic dataset1	2	1010	1	
Synthetic dataset2	2	2210	2	
Synthetic dataset3	2	3810	3	
Breast Cancer	32	357	0	benign
		212	1	malignant
Landsat Satellite	36	1072	1	red soi
		480	2	cotton crop
		961	3	grey soil
		415	4	damp grey soil
		370	5	soil with vegetation stubble
		1038	7	very damp grey soil
Glass	10	70	1	building windows float processed
		75	2	building windows non float processed
		17	3	vehicle windows float processed
		13	5	containers
		9	6	tableware
		28	7	headlamps

Table 4.1**:** The description of all datasets in experiments

of the table are the actual class label of all instances, and the columns are shown in Table 4.2.

The meaning of the true positive (TP), the false negative (FN), the false positive (FP) and the true negative (TN) in the confusion matrix are followed:

- True Positive (TP) is the number of the instances that are predicted as outliers, and they are actual outliers.
- False Negative (FN) is the number of the instances that are predicted as normal instances, but they are outliers.
- False Positive (FP) is the number of the instances that are predicted as outliers, but they are normal instances.
- True Negative (TN) is the number of the instances that are predicted as normal instances, and they are actual normal instances.

		predicted	
		outlier (+)	normal (-)
Actual	outlier (+)	True positive (TP)	False negative (FN)
	normal (-)	False positive (FP)	True negative (TN)

Table 4.2**:** Confusion Matrix

$$\text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.1}$$

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4.2}$$

To generate the ROC curve, outlier scores from all instances in the dataset are ranked in descending order. The ROC curve uses multiple cut-offs to generate the point on the ROC curve using TPR and FPR. An instance having the score above the cut-off will be identified as the outlier while the instance having the score less than or equal to the cut-off will be identified as the normal instance. The first cut-off for the first point is set at the maximum score and the other cut-offs uses the decreasing outlier scores. The best prediction would generate the graph from the lower left corner (0, 0) to the upper left corner (0, 1) and end at the upper right corner (1, 1). Therefore, the closer of the ROC curve to the upper left corner, the better the algorithm that generates the scores is.

Example 4.1**.** Dataset A contains 7 instances having one majority group and a three outliers as $A = \{(1, 5), (2, 4), (2, 5), (3, 6), (5, 4), (9, 13)\}$.

WOFs are computed and shown in Table 4.3.

First, the outlier scores are ranked in descending order. The calculation of the True Positive Rate (TPR) and the False Positive Rate (FPR) is performed using the cut-off by the decreasing outlier score as in Table 4.4. Finally, the ROC curve is plot in Figure 4.1.

instance	WOF score
(1,5)	0.6081
(2,4)	0.5440
(2,5)	0.7313
(3,6)	1.7123
(5,4)	3.0905
(9,13)	9.1253

Table 4.3: The WOF score for dataset A

instance	WOF score	Predicted	Actual
(9,13)	9.1253	outlier	outlier
(5,4)	3.0905	outlier	outlier
(3,6)	1.7123	normal	outlier
(2,5)	0.7313	normal	normal
(1,5)	0.6081	normal	normal
(2,4)	0.5440	normal	normal

Table 4.4: The ROC point at $(0, \frac{2}{3})$ using the cut-off = 1.7123

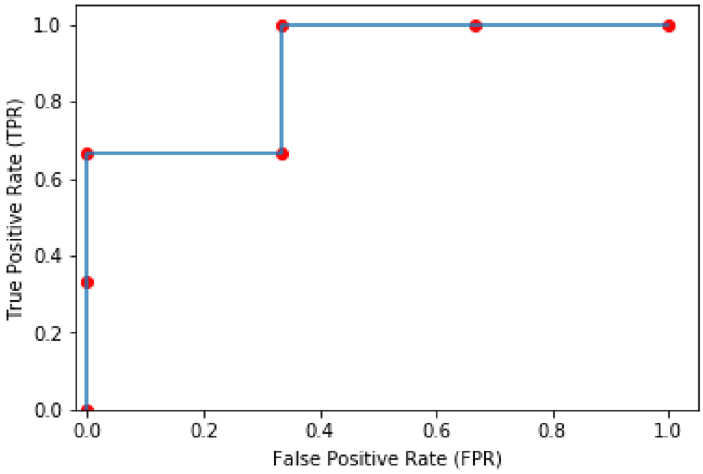


Figure 4.1: The ROC curve use WOF algorithm

The ROC curves contain useful information for understanding the accuracy. However, when the curves overlap, it is hard to recognize the best algorithm. Therefore, researchers use the area under the receiver operating characteristic curve (AUC) [17, 18] to obtain the best algorithm. AUC should be between 0.5 and 1.0. If it reaches 1, the algorithm perfectly separates instances into outliers and normal instances.

Wilcoxon signed rank test [19, 20] was used to compare the result performance of the algorithms. It is a non-parametric statistical procedure for comparing two samples.

This method is suitable for the samples which are relatively small and do not have the normal distribution.

Let k be the number of paired comparisons and S_i be the rank of the difference value of two techniques, where $i = 1, 2, 3, ..., k$. First, the difference between each sample pair is computed. Hence, the absolute values of the differences are the rank. For this method, the differences of zero are ignored when ranking. Then, the sum of ranks with positive differences denoted by R^+ and the sum of ranks with negative differences denoted by R^- are computed as the equation below.

$$R^+ = \sum_{s_i > 0} rank(S_i), \quad R^- = \sum_{s_i < 0} rank(S_i)$$

Let T be the smaller of sum, $T = min\{R^+, R^-\}$, the statistic

$$z = \frac{T - k(k+1)/4}{\sqrt{k(k+1)(2k+1)/24}}$$

for the larger number of the dataset [21]. The critical values for T can be found in tables published in the statistical textbooks if the number of datasets less than thirty. Table 4.5 shows the critical values of Wilcoxon signed rank test up to thirteen datasets.

k	Two-Tailed Test		One-Tailed Test	
	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$
5	–	–	0	–
6	0	–	2	–
7	2	–	3	0
8	3	0	5	1
9	5	1	8	3
10	8	3	10	5
11	10	5	13	7
12	13	7	17	9
13	17	9	21	12

Table 4.5**:** The critical values of Wilcoxon Signed Rank Test

The null hypotheses and alternative hypotheses for two-tailed test are set as follows:

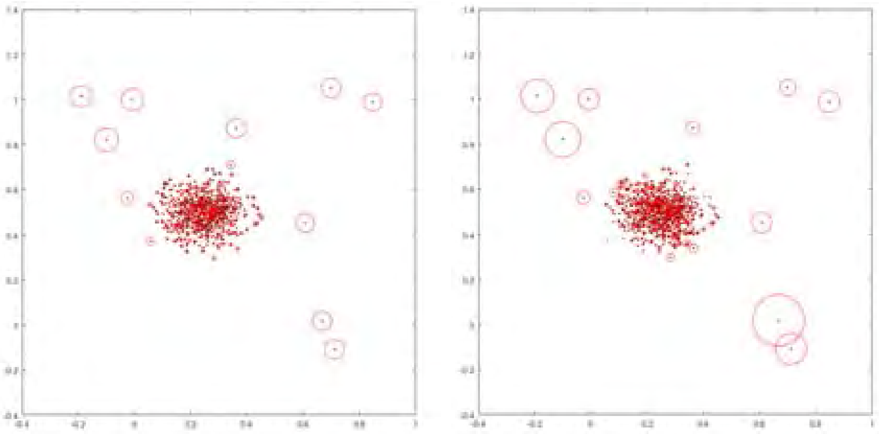
$$H_0 : \mu_1 - \mu_2 = 0,$$

$$H_1 : \mu_1 - \mu_2 \neq 0,$$

where μ_1 is the the area under the curve (AUC) of the WOF algorithm and μ_2 is the the area under the curve (AUC) of the OOF algorithm.

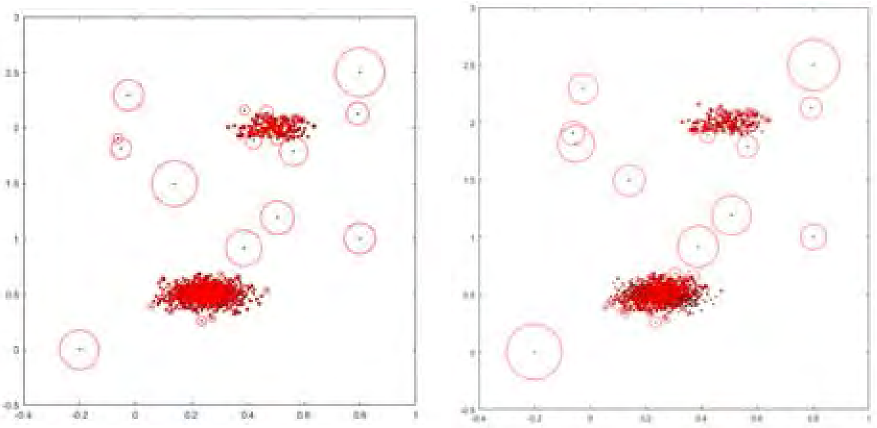
4.1.3 **A synthetic example**

Three collections of synthetic two-dimensional datasets are simulated for testing the WOF algorithm. The first collection contains ten datasets having a cluster of the normal distribution with ten outliers. The second collection contains eight datasets having two clusters of the normal distributions with ten outliers. The third collection contains ten datasets having two clusters of the normal distribution and one cluster of the uniform with ten outliers. Figure 4.2 shows one of all scatter plot of the first synthetic collection that contains 1010 instances where 1000 instances are in a cluster and 10 instances are outliers. The figure illustrates the result of OOFs and WOFs where the score is represented by the red circle. A large circle implies that instance is an outlier, while a small circle implies that instance is normal. The scatter plot in Figure 4.3 shows the result of WOFs on the second synthetic collection that contains 2210 instances where 2000 instances are in one cluster, 200 instances are sparse in another cluster, and 10 instances are outliers. The scatter plot in Figure 4.4 shows the result of WOFs on the third synthetic collection that contains 3810 instances where 3000 instances are in the ellipse cluster, 700 instances are in the square cluster, 100 instances are in the sparse cluster, and 10 instances are outliers.



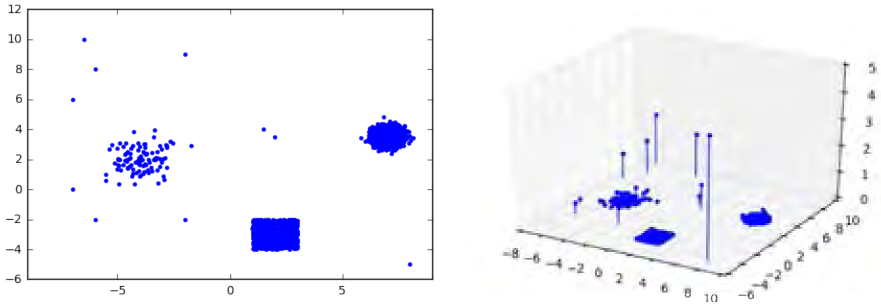
(a) The OOFs **(b)** The WOFs

Figure 4.2: Comparison of the OOF and the WOF on 1010 data points where the radius of the circle represents its score



(a) The OOFs **(b)** The WOFs

Figure 4.3: Comparison of the OOF and the WOF on 2210 data points where the radius of the circle represents its score



(a) A synthetics example **(b)** The WOFs

Figure 4.4: The WOFs on 3710 data points where the height represents its score

4.1.4 UCI datasets

Wisconsin Diagnostic Breast Cancer dataset

Wisconsin diagnostic breast cancer dataset has 569 instances, 32 attributes in 2 classes; the class “benign” has 357 instances and the class “malignant” has 212 instances. The class “benign” chosen as the majority data instances in the Wisconsin diagnostic breast cancer dataset and 10 instances from malignant instances are picked randomly. The generated datasets from Wisconsin diagnostic breast cancer dataset are performed ten times for testing.

Figure 4.5 - 4.14 and Table 4.6 - 4.15 show the ROC curves and top-10 ranks of Wisconsin diagnostic breast cancer dataset (Bcd) between WOFs and OOFs, sorted by its score. The bold numbers are marked when OOF and WOF identify these instances as outlier, correctly. Note that the AUC of the WOF algorithm is better than the AUC of the OOF algorithm.

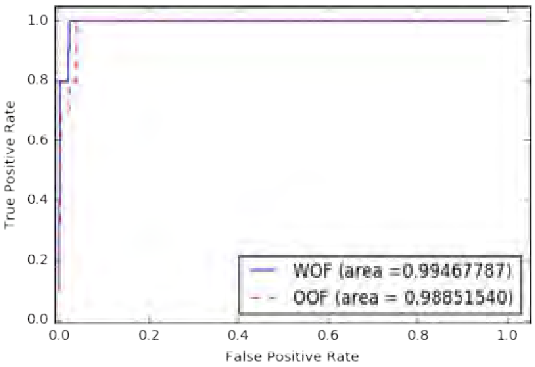


Figure 4.5**:** The ROC between WOFs and OOFs for Bcd 1

Rank	OOF		WOF	
	index	score	index	score
1	**367**	**438.471**	**367**	**456.368**
2	**359**	**195.501**	**359**	**289.572**
3	**361**	**136.928**	**361**	**164.978**
4	300	95.883	**362**	**116.515**
5	**364**	**74.257**	300	107.98
6	**363**	**63.687**	**366**	**98.356**
7	**366**	**58.67**	**363**	**81.871**
8	74	30.947	**364**	**79.039**
9	**360**	**23.104**	**360**	**50.458**
10	203	22.523	36	48.564

Table 4.6**:** The score for Bcd 1

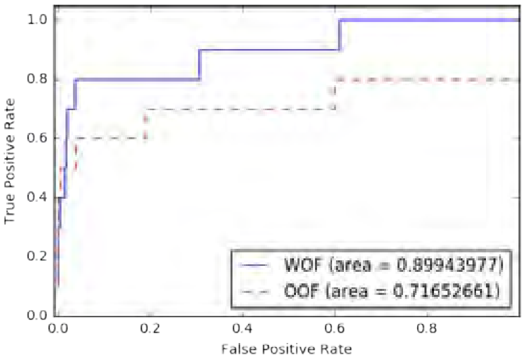


Figure 4.6**:** The ROC between WOFs and OOFs for Bcd 2

Rank	OOF		WOF	
	index	score	index	score
1	**361**	**345.109**	**361**	**380.472**
2	**362**	**65.068**	**363**	**113.352**
3	**363**	**63.711**	**362**	**72.569**
4	**365**	**49.814**	203	71.522
5	203	31.614	**365**	**49.618**
6	74	30.691	36	48.775
7	**366**	**29.862**	74	46.461
8	36	17.773	300	45.006
9	208	16.919	312	38.421
10	312	16.529	**366**	**35.692**

Table 4.7**:** The score for Bcd 2

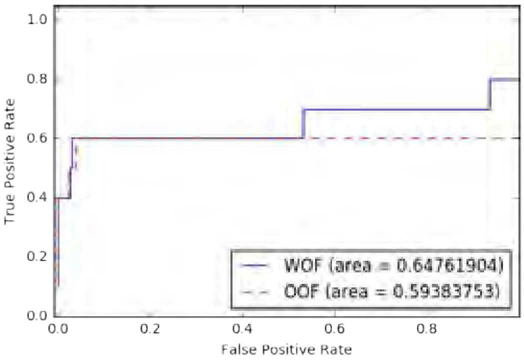


Figure 4.7: The ROC between WOFs and OOFs for Bcd 3

Rank	OOF		WOF	
	index	score	index	score
1	**358**	**1230.5**	**358**	**1276.322**
2	**359**	**304.783**	**359**	**372.419**
3	**360**	**214.89**	**360**	**233.234**
4	**365**	**97.921**	**365**	**141.176**
5	300	55.938	300	69.672
6	203	31.324	203	60.094
7	74	30.685	36	48.051
8	36	19.573	74	44.051
9	312	16.461	312	34.986
10	334	11.982	334	25.296

Table 4.8: The score for Bcd 3

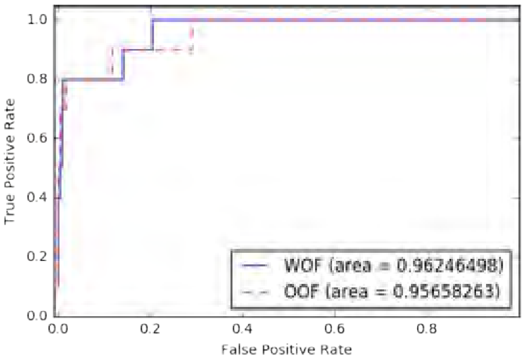


Figure 4.8: The ROC between WOFs and OOFs for Bcd 4

Rank	OOF		WOF	
	index	score	index	score
1	**362**	**450.998**	**362**	**620.831**
2	**359**	**216.197**	**366**	**451.732**
3	**360**	**172.097**	**359**	**378.512**
4	**366**	**140.417**	**360**	**219.271**
5	**364**	**137.15**	300	172.674
6	300	131.478	**364**	**139.853**
7	**361**	**33.537**	203	70.689
8	203	31.735	36	48.73
9	**363**	**23.786**	**363**	**42.558**
10	36	18.86	**361**	**40.114**

Table 4.9: The score for Bcd 4

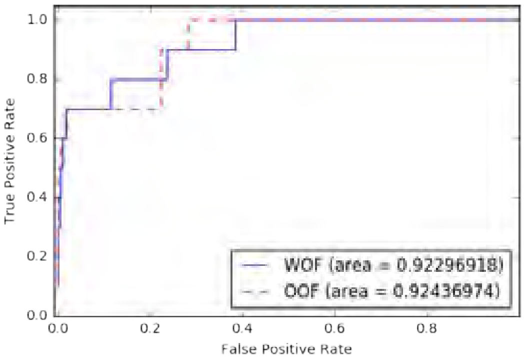


Figure 4.9: The ROC between WOFs and OOFs for Bcd 5

Rank	OOF		WOF	
	index	score	index	score
1	**358**	**67.352**	**361**	**436.078**
2	**361**	**67.117**	**366**	**166.417**
3	**366**	**63.485**	**364**	**108.488**
4	**362**	**49.241**	300	104.451
5	**364**	**37.228**	**358**	**82.562**
6	203	31.508	203	70.668
7	36	20.312	**362**	**55.57**
8	**367**	**19.894**	36	48.807
9	208	17.084	312	34.59
10	312	16.589	**367**	**26.117**

Table 4.10: The score for Bcd 5

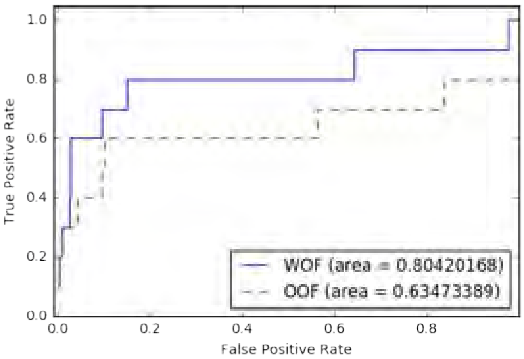


Figure 4.10**:** The ROC between WOFs and OOFs for Bcd 6

Rank	OOF		WOF	
	index	score	index	score
1	**360**	**399.773**	**360**	**441.318**
2	300	75.146	300	180.365
3	**366**	**63.501**	**366**	**80.793**
4	203	31.358	203	71.914
5	74	30.39	36	48.856
6	36	19.188	74	43.636
7	**364**	**18.895**	**364**	**29.637**
8	208	15.224	334	24.968
9	334	11.98	208	23.324
10	295	11.285	295	20.177

Table 4.11**:** The score for Bcd 6

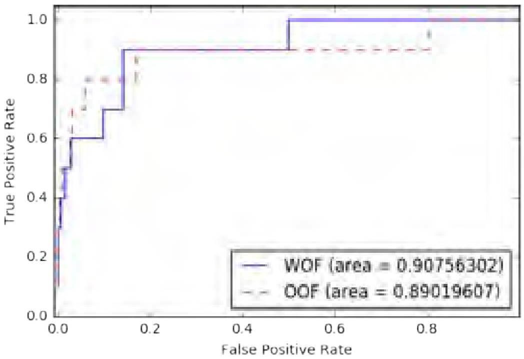


Figure 4.11**:** The ROC between WOFs and OOFs for Bcd 7

Rank	OOF		WOF	
	index	score	index	score
1	**358**	**874.095**	**358**	**906.865**
2	**366**	**122.042**	**366**	**165.588**
3	**362**	**53.838**	**361**	**103.838**
4	300	46.68	300	87.956
5	**359**	**34.633**	203	71.362
6	203	31.592	**362**	**69.477**
7	74	30.911	36	48.663
8	**361**	**19.589**	74	47.226
9	36	17.625	312	35.357
10	312	16.315	**359**	**32.21**

Table 4.12**:** The score for Bcd 7

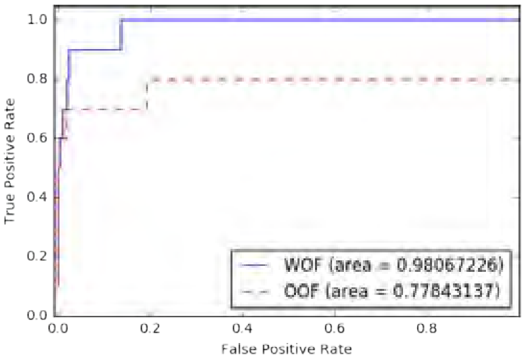


Figure 4.12**:** The ROC between WOFs and OOFs for Bcd 8

Rank	OOF		WOF	
	index	score	index	score
1	**359**	**505.804**	**359**	**518.662**
2	**362**	**174.574**	**361**	**283.205**
3	**366**	**166.772**	**366**	**256.763**
4	**367**	**145.836**	**362**	**195.326**
5	**363**	**137.459**	**367**	**172.628**
6	203	28.537	300	110.893
7	**361**	**21.28**	**363**	**107.221**
8	36	19.543	36	48.606
9	208	16.983	203	38.928
10	312	16.691	312	32.278

Table 4.13**:** The score for Bcd 8

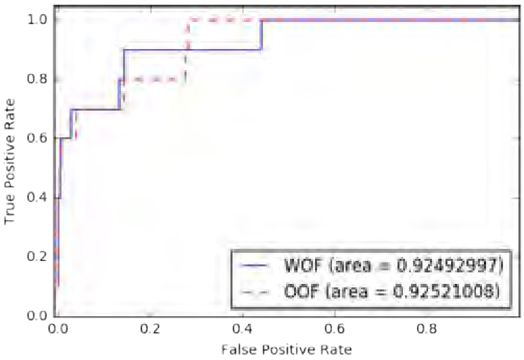


Figure 4.13: The ROC between WOFs and OOFs for Bcd 9

Rank	OOF		WOF	
	index	score	index	score
1	**358**	**339.824**	**358**	**450.122**
2	**361**	**243.504**	**361**	**224.599**
3	**364**	**152.318**	**364**	**115.758**
4	**366**	**98.204**	**360**	**111.017**
5	300	55.483	300	107.386
6	**367**	**34.275**	**366**	**90.048**
7	74	30.486	36	48.052
8	**360**	**26.195**	**367**	**47.052**
9	36	19.116	203	37.993
10	208	16.799	74	36.393

Table 4.14: The score for Bcd 9

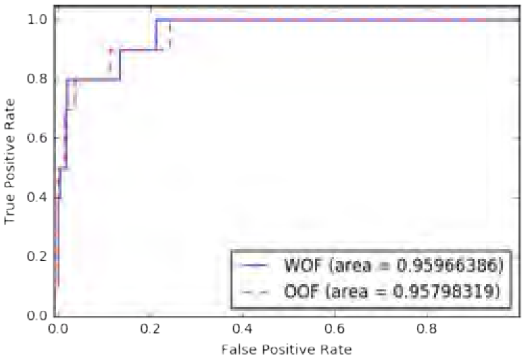


Figure 4.14: The ROC between WOFs and OOFs for Bcd 10

Rank	OOF		WOF	
	index	score	index	score
1	**358**	**550.652**	**358**	**565.656**
2	**363**	**177.492**	**359**	**290.455**
3	**359**	**167.768**	**363**	**188.925**
4	**366**	**63.941**	**366**	**124.656**
5	**365**	**31.41**	203	67.972
6	203	31.35	**365**	**51.517**
7	36	18.15	36	48.731
8	300	17.319	300	33.51
9	74	16.887	208	31.141
10	208	16.835	74	29.618

Table 4.15: The score for Bcd 10

Table 4.16 shows the significant test of the AUC performance between the WOF algorithm and the OOF algorithm. Note, this table, “Bcd i” represents Wisconsin diagnostic breast cancer dataset i when $i = 1, 2, ..., 10$. The ranks are assigned from the lowest to the highest of the absolute difference. The sum of all ranks for the positive difference is $R^+ = 5 + 9 + 7 + 4 + 8 + 6 + 10 + 3 = 47$ and the sum of all ranks for the negative difference is $R^- = 2 + 1 = 3$. According to Table 4.5, a level of significance is $\alpha = 0.05$ and $n = 10$ datasets. $T = min\{47, 3\} = 3$, which is less than the critical value for Wilcoxon signed rank test (8) so it rejects the null-hypothesis.

Dataset	the WOF algorithm	the OOF algorithm	Difference	Rank
Bcd 1	0.99467787	0.98851540	+0.00616246	5
Bcd 2	0.89943977	0.71652661	+0.18291316	9
Bcd 3	0.64761904	0.59383753	+0.05378151	7
Bcd 4	0.96246498	0.95658263	+0.00588235	4
Bcd 5	0.92296918	0.92436974	-0.00140056	2
Bcd 6	0.80420168	0.63473389	+0.16946779	8
Bcd 7	0.90756302	0.89019607	+0.01736694	6
Bcd 8	0.98067226	0.77843137	+0.20224089	10
Bcd 9	0.92492997	0.92521008	-0.00028011	1
Bcd 10	0.95966386	0.95798319	+0.00168067	3
$R^{+} = 47,\ R^{-} = 3$				

Table 4.16: The significant test of the AUC performance between the WOF algorithm and the OOF algorithm

Statlog (Landsat Satellite) dataset

Statlog (Landsat Satellite) dataset has 4435 instances, 36 attributes in 6 classes. The class “1: red soil” chosen as the majority data instances in the statlog (Landsat Satellite) dataset and 10 instances from the other classes are picked randomly. The generated datasets from statlog (Landsat Satellite) dataset are performed ten times for testing.

Figure 4.15 - 4.24 and Table 4.17 - 4.26 show ROC and top-10 ranks of the statlog (Landsat Satellite) dataset between WOFs and OOFs, sorted by its score. The bold numbers are marked when OOF and WOF identify these instances as outlier, correctly. Note that the AUC of the WOF algorithm is better than the AUC of the OOF algorithm.

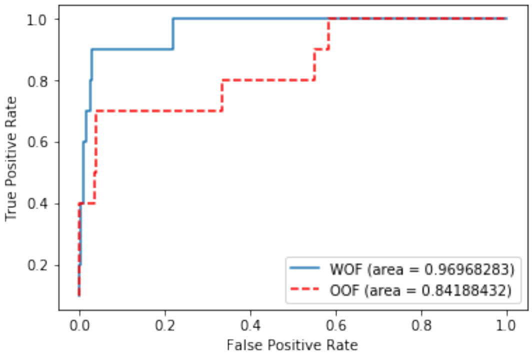


Figure 4.15: The ROC between WOFs and OOFs for Statlog 1

Rank	OOF		WOF	
	index	score	index	score
1	**1079**	**55.367**	**1079**	**132.469**
2	**1082**	**14.406**	**1077**	**39.964**
3	**1077**	**8.9442**	714	12.947
4	**1073**	**4.848**	**1074**	**3.774**
5	622	3.747	**1081**	**3.756**
6	591	1.968	840	2.509
7	720	1.22	246	2.251
8	687	1.187	839	2.198
9	621	1.096	613	2.141
10	688	1.064	773	2.102

Table 4.17: The score for Statlog 1

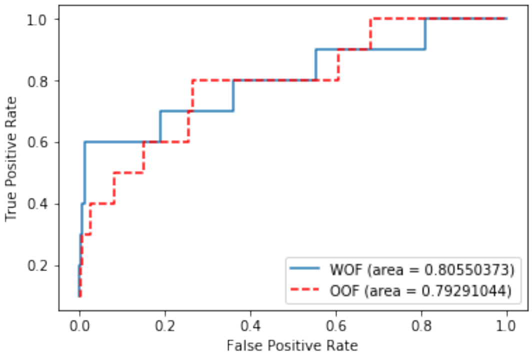


Figure 4.16: The ROC between WOFs and OOFs for Statlog 2

Rank	OOF		WOF	
	index	score	index	score
1	**1081**	**16.156**	**1081**	**82.268**
2	622	3.412	**1077**	**17.503**
3	**1079**	**3.088**	613	9.007
4	591	1.546	584	5.33
5	720	1.219	372	3.547
6	687	1.117	**1076**	**3.542**
7	**1080**	**1.112**	622	2.749
8	688	1.06	246	1.847
9	620	1.03	222	1.679
10	653	1.027	295	1.667

Table 4.18: The score for Statlog 2

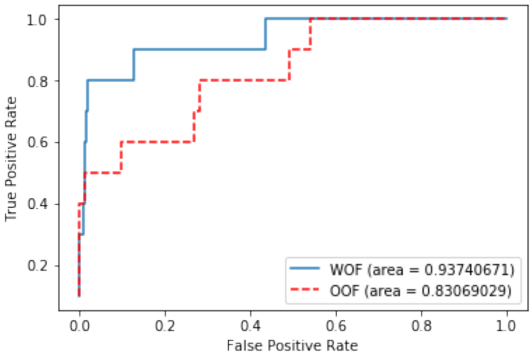


Figure 4.17: The ROC between WOFs and OOFs for Statlog 3

Rank	OOF		WOF	
	index	score	index	score
1	**1079**	**34.319**	**1079**	**104.952**
2	**1073**	**29.763**	**1073**	**61.845**
3	**1080**	**12.496**	**1080**	**24.698**
4	**1078**	**4.108**	714	13.345
5	622	2.873	840	3.776
6	591	1.947	584	3.197
7	687	1.172	622	2.321
8	720	1.105	222	1.841
9	621	1.083	**1072**	**1.786**
10	1024	1.062	839	1.772

Table 4.19: The score for Statlog 3

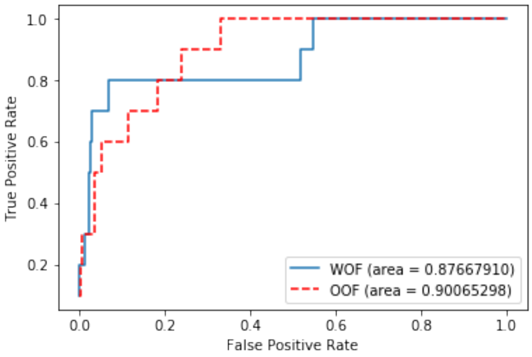


Figure 4.18: The ROC between WOFs and OOFs for Statlog 4

Rank	OOF		WOF	
	index	score	index	score
1	**1074**	**60.138**	**1074**	**115.093**
2	622	2.648	**1081**	**28.207**
3	591	1.859	714	11.617
4	**1082**	**1.632**	584	3.019
5	720	1.193	808	2.728
6	687	1.17	622	2.421
7	**1081**	**1.119**	222	2.27
8	688	1.063	246	1.978
9	653	1.047	591	1.771
10	620	1.026	688	1.619

Table 4.20: The score for Statlog 4

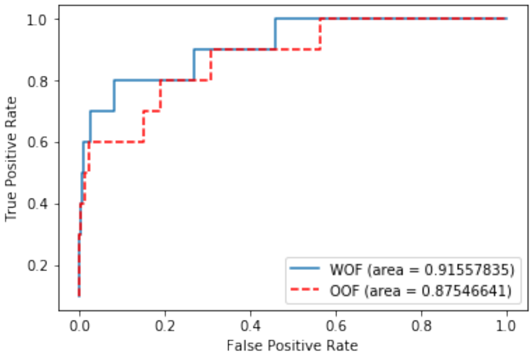


Figure 4.19: The ROC between WOFs and OOFs for Statlog 5

Rank	OOF		WOF	
	index	score	index	score
1	**1081**	**32.024**	**1081**	**82.52**
2	**1073**	**11.174**	**1077**	**36.354**
3	**1080**	**3.998**	**1073**	**31.521**
4	622	3.139	714	10.49
5	**1077**	**2.76**	**1080**	**9.474**
6	591	1.938	840	4.475
7	687	1.171	247	2.405
8	720	1.065	773	2.005
9	621	1.024	**1082**	**1.886**
10	1024	1.01	829	1.649

Table 4.21: The score for Statlog 5

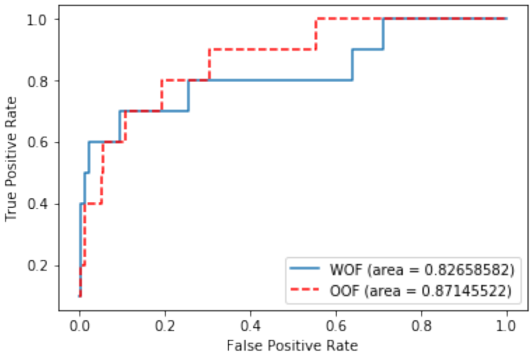


Figure 4.20: The ROC between WOFs and OOFs for Statlog 6

Rank	OOF		WOF	
	index	score	index	score
1	**1073**	**7.327**	**1073**	**17.703**
2	622	3.502	12	9.27
3	591	1.532	**1082**	**8.571**
4	**1076**	**1.182**	**1081**	**8.158**
5	687	1.107	622	3.95
6	1024	1.035	11	3.589
7	720	1.035	**1076**	**2.531**
8	12	0.978	591	2.197
9	621	0.937	688	1.862
10	246	0.908	687	1.269

Table 4.22: The score for Statlog 6

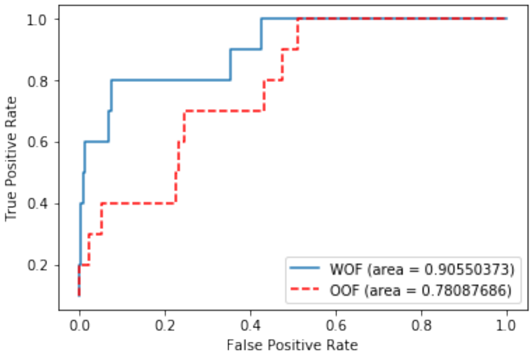


Figure 4.21: The ROC between WOFs and OOFs for Statlog 7

Rank	OOF		WOF	
	index	score	index	score
1	**1078**	**32.969**	**1078**	**104.426**
2	**1082**	**10.172**	**1080**	**16.983**
3	622	3.748	714	11.737
4	591	1.977	**1076**	**7.102**
5	720	1.238	840	3.443
6	687	1.193	246	3.398
7	621	1.093	**1075**	**2.488**
8	688	1.072	591	1.718
9	653	1.055	687	1.499
10	620	1.039	688	1.463

Table 4.23: The score for Statlog 7

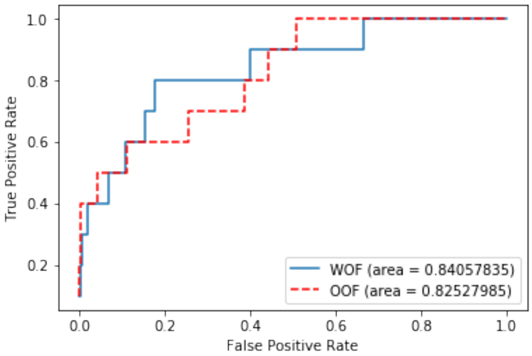


Figure 4.22: The ROC between WOFs and OOFs for Statlog 8

Rank	OOF		WOF	
	index	score	index	score
1	**1082**	**26.781**	**1082**	**50.303**
2	**1076**	**7.42**	12	9.385
3	622	2.489	**1076**	**7.869**
4	**1075**	**1.967**	622	6.26
5	591	1.826	11	3.787
6	**1074**	**1.443**	653	2.218
7	720	1.17	621	2.215
8	687	1.134	720	2.135
9	1024	1.024	688	2.025
10	653	1.004	**1075**	**1.831**

Table 4.24: The score for Statlog 8

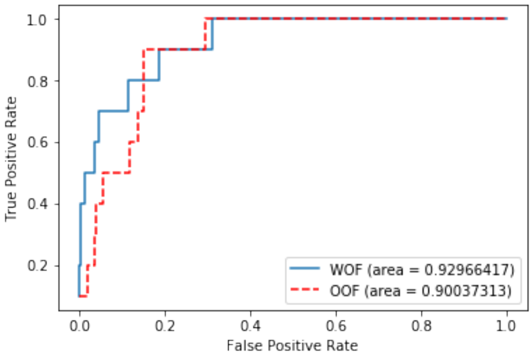


Figure 4.23: The ROC between WOFs and OOFs for Statlog 9

Rank	OOF		WOF	
	index	score	index	score
1	**1082**	**63.815**	**1082**	**129.786**
2	622	3.733	**1078**	**25.65**
3	591	1.961	714	12.26
4	687	1.184	**1076**	**4.116**
5	720	1.158	**1073**	**3.195**
6	621	1.084	613	2.35
7	1024	1.048	622	1.977
8	653	1.041	808	1.889
9	688	1.037	840	1.862
10	620	1.019	591	1.766

Table 4.25: The score for Statlog 9

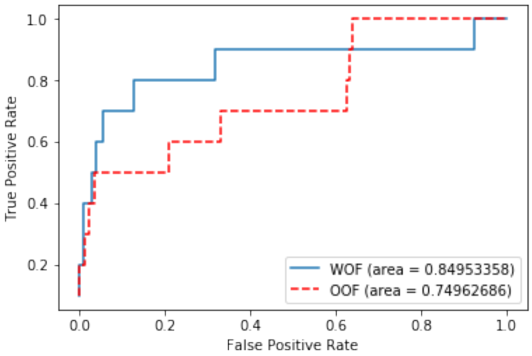


Figure 4.24: The ROC between WOFs and OOFs for Statlog 10

Rank	OOF		WOF	
	index	score	index	score
1	**1081**	**53.896**	**1081**	**129.791**
2	**1082**	**14.707**	**1077**	**34.523**
3	622	3.714	714	12.016
4	591	1.943	840	4.146
5	720	1.179	773	2.316
6	687	1.152	222	1.742
7	621	1.087	246	1.661
8	1024	1.005	12	1.651
9	688	0.969	808	1.474
10	246	0.94	**1072**	**1.389**

Table 4.26: The score for Statlog 10

Table 4.27 shows the significant test of the AUC performance between the WOF algorithm and the OOF algorithm. “Statlog i” represents the statlog (Landsat Satellite) dataset i when $i = 1, 2, ..., 10$. The ranks are assigned from the lowest to the highest of the absolute difference. The sum of all ranks for the positive difference is $R^+ = 10 + 1 + 8 + 5 + 9 + 2 + 4 + 7 = 41$ and the sum of all ranks for the negative difference is $R^{-} = 3 + 6 = 9$. According to Table 4.5, a level of significance is $\alpha = 0.05$ and $n = 10$ datasets. $T = min\{41, 9\} = 9$, which is more than the critical value for the Wilcoxon signed rank test (8) so it fails to reject the null-hypothesis.

Dataset	WOF algorithm	OOF algorithm	Difference	Rank
Statlog 1	0.96968283	0.84188432	+0.12779850	10
Statlog 2	0.80550373	0.79291044	+0.01259328	1
Statlog 3	0.93740671	0.83069029	+0.10671641	8
Statlog 4	0.87667910	0.90065298	-0.02397388	3
Statlog 5	0.91557835	0.87546641	+0.04011193	5
Statlog 6	0.82658582	0.87145522	-0.04486939	6
Statlog 7	0.90550373	0.78087686	+0.12462686	9
Statlog 8	0.84057835	0.82527985	+0.01529849	2
Statlog 9	0.92966417	0.90037313	+0.02929104	4
Statlog 10	0.84953358	0.74962686	+0.09990672	7
		$R^{+} = 41$, $R^{-} = 9$		

Table 4.27: The significant test of average AUC performance between the WOF algorithm and the OOF algorithm

Glass identification dataset

Glass identification dataset has 214 instances, 10 attributes in 6 classes. The class “2 : building windows non float processed” chosen as the majority data instances in the glass identification dataset and 10 instances from the other class are picked randomly. The generated datasets from glass identification dataset are performed ten times for testing.

Figure 4.15 - 4.24 and Table 4.17 - 4.26 show the ROC curves and top-10 ranks of the glass identification dataset between WOFs and OOFs, sorted by its score. The bold numbers are marked when OOF and WOF identify these instances as outlier, correctly. Note the AUC of the WOF algorithm is better then the AUC of the OOF algorithm.

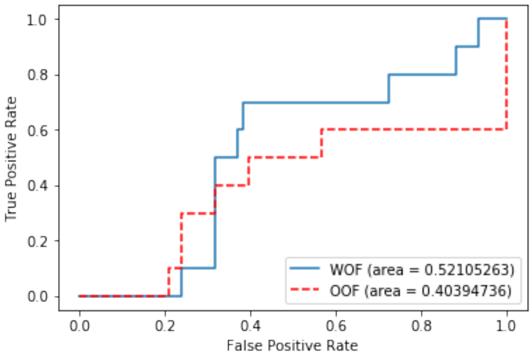


Figure 4.25: The ROC between WOFs and OOFs for Glass 1

Rank	OOF		WOF	
	index	score	index	score
1	38	0.976	38	1.765
2	36	0.852	62	0.778
3	43	0.761	43	0.759
4	35	0.537	37	0.733
5	39	0.499	15	0.671
6	62	0.485	36	0.544
7	40	0.389	34	0.475
8	61	0.367	35	0.32
9	41	0.318	61	0.294
10	34	0.285	60	0.291

Table 4.28: The score for Glass 1

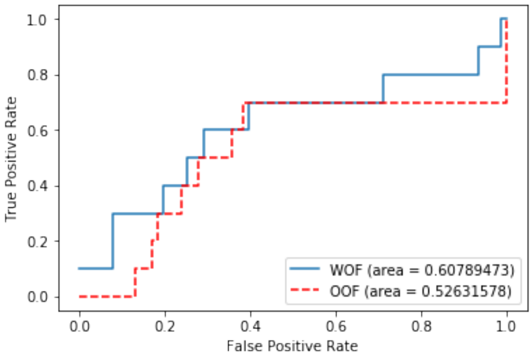


Figure 4.26: The ROC between WOFs and OOFs for Glass 2

Rank	OOF		WOF	
	index	score	index	score
1	38	0.944	**85**	**3.632**
2	36	0.779	38	1.543
3	35	0.58	37	0.695
4	43	0.505	43	0.653
5	62	0.397	36	0.602
6	39	0.338	41	0.42
7	41	0.282	35	0.306
8	61	0.273	**79**	**0.275**
9	60	0.268	**84**	**0.266**
10	40	0.263	34	0.253

Table 4.29: The score for Glass 2

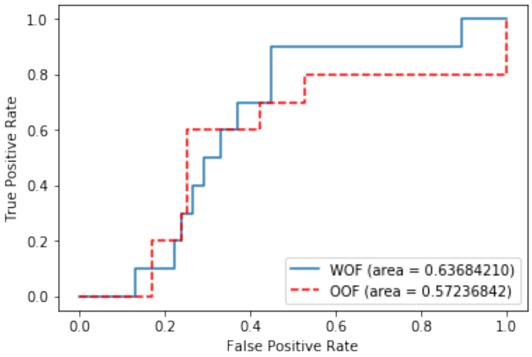


Figure 4.27: The ROC between WOFs and OOFs for Glass 3

Rank	OOF		WOF	
	index	score	index	score
1	38	0.971	38	1.765
2	36	0.846	62	0.778
3	43	0.67	43	0.759
4	35	0.541	37	0.733
5	62	0.449	15	0.671
6	39	0.333	36	0.544
7	40	0.317	40	0.286
8	41	0.314	34	0.282
9	60	0.279	61	0.277
10	61	0.271	60	0.276

Table 4.30: The score for Glass 3

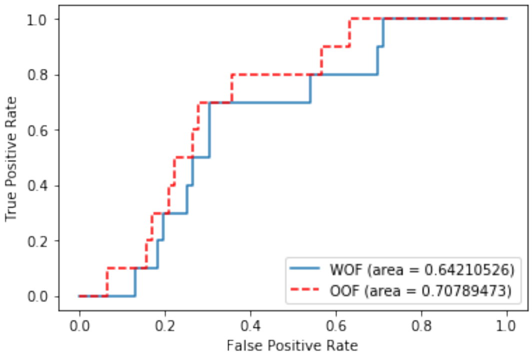


Figure 4.28: The ROC between WOFs and OOFs for Glass 4

Rank	OOF		WOF	
	index	score	index	score
1	38	0.96	38	1.765
2	36	0.846	62	0.778
3	43	0.665	43	0.759
4	35	0.543	37	0.733
5	62	0.437	15	0.671
6	**86**	**0.314**	36	0.544
7	41	0.313	40	0.288
8	60	0.254	35	0.281
9	61	0.236	59	0.271
10	34	0.222	61	0.258

Table 4.31: The score for Glass 4

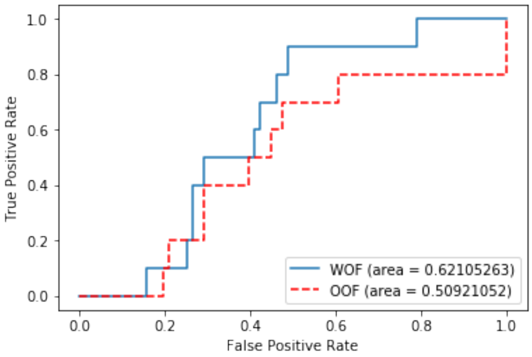


Figure 4.29: The ROC between WOFs and OOFs for Glass 5

Rank	OOF		WOF	
	index	score	index	score
1	38	0.981	38	1.765
2	36	0.818	62	0.778
3	43	0.681	43	0.759
4	35	0.575	37	0.733
5	62	0.509	15	0.671
6	39	0.383	36	0.544
7	40	0.323	34	0.475
8	41	0.32	35	0.301
9	61	0.295	61	0.294
10	60	0.263	40	0.285

Table 4.32: The score for Glass 5

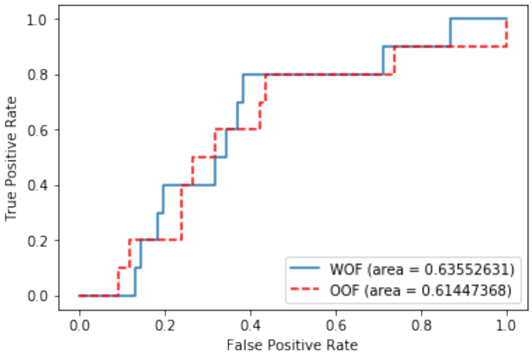


Figure 4.30: The ROC between WOFs and OOFs for Glass 6

Rank	OOF		WOF	
	index	score	index	score
1	38	0.961	38	1.765
2	36	0.868	43	0.759
3	43	0.66	37	0.733
4	35	0.516	62	0.721
5	62	0.468	15	0.671
6	39	0.326	36	0.544
7	41	0.316	34	0.379
8	**83**	**0.31**	35	0.321
9	40	0.292	61	0.294
10	60	0.281	39	0.254

Table 4.33: The score for Glass 6

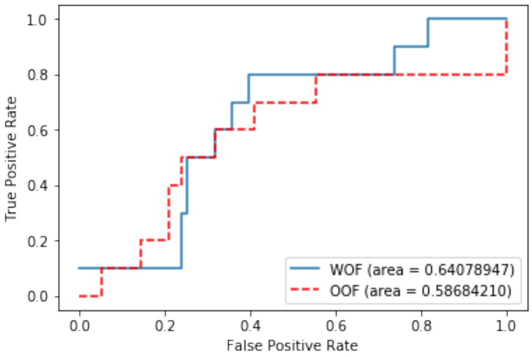


Figure 4.31: The ROC between WOFs and OOFs for Glass 7

Rank	OOF		WOF	
	index	score	index	score
1	38	0.977	**85**	**1.983**
2	36	0.831	38	1.894
3	43	0.659	36	1.186
4	62	0.483	37	0.868
5	**85**	**0.454**	34	0.492
6	39	0.356	15	0.429
7	35	0.343	43	0.421
8	41	0.321	41	0.366
9	40	0.296	60	0.344
10	61	0.254	59	0.343

Table 4.34: The score for Glass 7

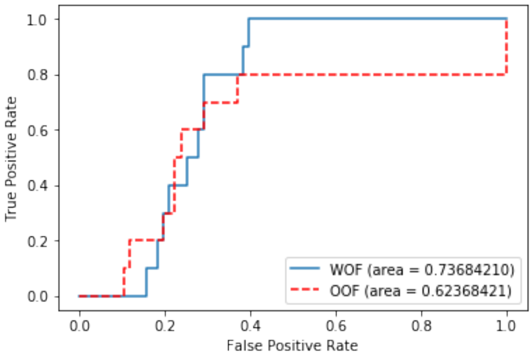


Figure 4.32: The ROC between WOFs and OOFs for Glass 8

Rank	OOF		WOF	
	index	score	index	score
1	38	0.935	38	1.765
2	36	0.644	43	0.759
3	43	0.64	37	0.7
4	35	0.553	15	0.671
5	62	0.417	62	0.609
6	61	0.349	36	0.508
7	39	0.342	35	0.322
8	41	0.308	34	0.32
9	**86**	**0.269**	61	0.293
10	40	0.268	39	0.256

Table 4.35: The score for Glass 8

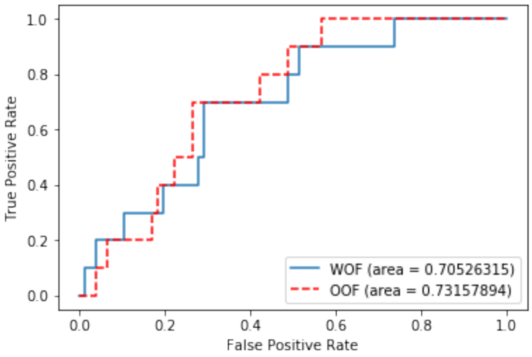


Figure 4.33: The ROC between WOFs and OOFs for Glass 9

Rank	OOF		WOF	
	index	score	index	score
1	38	0.945	38	1.894
2	36	0.839	**78**	**1.406**
3	43	0.518	36	1.186
4	**78**	**0.485**	37	0.868
5	62	0.46	**83**	**0.791**
6	39	0.44	34	0.492
7	**83**	**0.378**	15	0.429
8	35	0.363	43	0.421
9	40	0.352	41	0.366
10	41	0.294	35	0.327

Table 4.36: The score for Glass 9

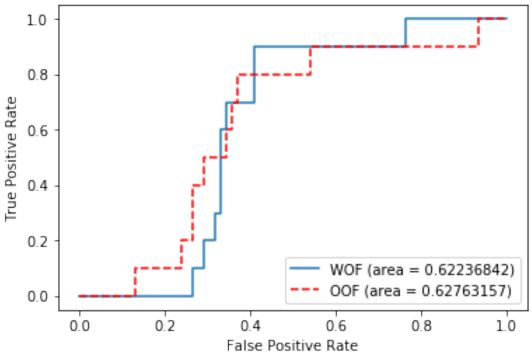


Figure 4.34**:** The ROC between WOFs and OOFs for Glass 10

Rank	OOF		WOF	
	index	score	index	score
1	38	0.988	38	1.765
2	36	0.832	62	0.778
3	43	0.681	43	0.759
4	35	0.554	37	0.733
5	62	0.495	15	0.671
6	39	0.364	36	0.544
7	40	0.328	34	0.475
8	61	0.325	59	0.315
9	41	0.314	61	0.294
10	60	0.287	60	0.289

Table 4.37**:** The score for Glass 10

Table 4.38 shows the significant test of the AUC performance between the WOF algorithm and the OOF algorithm. “Glass i” represents the glass identification dataset i when $i = 1, 2, ..., 10$. The ranks are assigned from the lowest to the highest of the absolute difference. The sum of all ranks for the positive difference is $R^+ = 10{+}7{+}5{+}8{+}2{+}4{+}9 = 40$ and the sum of all ranks for the negative difference is $R^- = 6{+}3{+}1 = 10$. According to Table 4.5, a level of significance is $\alpha = 0.05$ and $n = 10$ datasets. $T = min\{40, 10\} = 10$, which is more than the critical value for Wilcoxon signed rank test is (8) so it fails to reject the null-hypothesis.

Dataset	The WOF algorithm	The OOF algorithm	Difference	Rank
Glass 1	0.52105263	0.40394736	+0.11710527	10
Glass 2	0.60789473	0.52631578	+0.08157895	7
Glass 3	0.63684210	0.57236842	+0.06447367	5
Glass 4	0.64210526	0.70789473	-0.06578947	6
Glass 5	0.62105263	0.50921052	+0.11184211	8
Glass 6	0.63552631	0.61447368	+0.02105263	2
Glass 7	0.64078947	0.58684210	+0.05394736	4
Glass 8	0.73684210	0.62368421	+0.11315789	9
Glass 9	0.70526315	0.73157894	-0.02631579	3
Glass 10	0.62236842	0.62763157	-0.00526314	1
		$R^+ = 40,\ R^- = 10$		

Table 4.38**:** The significant test of the average AUC performance between the WOF algorithm and the OOF algorithm

4.2 The Efficiency of the WOF Algorithm

This section shows the running time comparison of the WOF algorithm and the

OOF algorithm. To compare the running time for computing the outlier score, the large size of the synthetic dataset are generated and implemented via Python programming language. Figure 4.35 shows the running time of the random synthetic datasets and varies the data from 100 instances to 20,000 instances. The WOF algorithm uses less running time than the OOF algorithm.

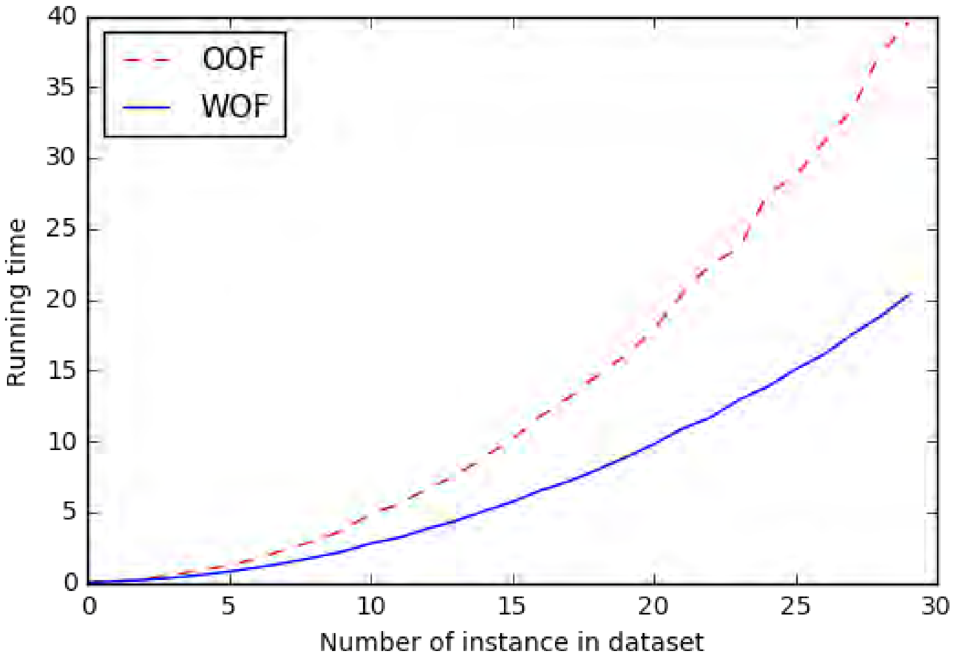


Figure 4.35: The running time of the synthetic dataset

CHAPTER V

CONCLUSIONS

The new algorithm to compute an outlier score for each instance call the weighted minimum consecutive pair of the extreme pole outlier factor (WOF) algorithm is presented. It is implemented using python language. The WOF algorithm does not require any parameter to compute WOFs for all instances. From the extreme pole of the dataset, WOF of each instance p is the weighted summation of the distance between p and its adjacent instances on the core vector. Since the core vector is generated from two extreme poles, WOF is set as the average computation from each extreme pole. If an instance has low WOF, then this instance is identified as the normal instance. If WOF of this instance is high, then it indicates that this instance is an outlier. From the experimental result of three synthetic datasets, it showed that the WOF algorithm can detect the same number of outliers as the OOF algorithm. Moreover, from the experimental results of three real world datasets, it showed that the WOF algorithm can detect more outliers than the OOF algorithm.

In terms of the time complexity between the WOF algorithm and the OOF algorithm, the WOF algorithm has $O(n^2)$ time complexity where n be the number of instances in a dataset. It is lower than the time complexity of the OOF algorithm which is $O(n^2 \log n)$.

However, the WOF algorithm has some weak points. If the distance between the outlier which is the radial projection on the core vector and the consecutive pair instances is close to the group, it is classified to be the normal. Then the distance between two instances is calculated from the other way such that using the real distance between the instance to the nearest instance.

Currently there are a lot of data or big data. For the future work, the WOF algorithm could be improved to run in linear time.

REFERENCES

[1] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[2] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[3] E. M. Knox and R. T. Ng, “Algorithms for mining distancebased outliers in large datasets,” in *Proceedings of the International Conference on Very Large Data Bases*, pp. 392–403, Citeseer, 1998.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *ACM sigmod record*, vol. 29, pp. 93–104, ACM, 2000.

[5] N. Buthong, A. Luangsodsai, and K. Sinapiromsaran, “Outlier detection score based on ordered distance difference,” in *Computer Science and Engineering Conference (ICSEC), 2013 International*, pp. 157–162, IEEE, 2013.

[6] X. Ru, Z. Liu, Z. Huang, and W. Jiang, “Normalized residual-based constant false-alarm rate outlier detection,” *Pattern Recognition Letters*, vol. 69, pp. 1–7, 2016.

[7] J. Tang, Z. Chen, A. Fu, and D. Cheung, “Enhancing effectiveness of outlier detections for low density patterns,” *Advances in Knowledge Discovery and Data Mining*, pp. 535–548, 2002.

[8] M. Goldstein and A. Dengel, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

[9] B. Schweizer and A. Sklar, “Statistical metric spaces,” *Pacific journal of mathematics*, vol. 10, no. 1, pp. 313–334, 1960.

[10] J. P. Van de Geer, *Some Aspects of Minkowski distance.* Leiden University, Department of Data Theory, 1995.

[11] S. Lele, "Euclidean distance matrix analysis (edma): estimation of mean form and mean form difference," *Mathematical Geology*, vol. 25, no. 5, pp. 573–602, 1993.

[12] B. Kaveelerdpotjana, K. Sinapiromsaran, and B. Intiyot, "Farthest boundary clustering algorithm: half-orbital extreme pole," in *Computer Science and Engineering Conference (ICSEC), 2013 International*, pp. 168–173, IEEE, 2013.

[13] K. Sinapiromsaran and N. Techaval, "Network intrusion detection using multi-attributed frame decision tree," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on*, pp. 203–207, IEEE, 2012.

[14] C.-Y. Chen, S.-C. Hwang, and Y.-J. Oyang, "A statistics-based approach to control the quality of subclusters in incremental gravitational clustering," *Pattern Recognition*, vol. 38, no. 12, pp. 2256–2269, 2005.

[15] I. SageMath, *CoCalc Collaborative Computation Online*, 2016. `https://cocalc.com/`.

[16] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[17] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine.," *Clinical chemistry*, vol. 39, no. 4, pp. 561–577, 1993.

[18] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[19] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[21] B. Trawiński, M. Smętek, Z. Telec, and T. Lasota, "Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 4, pp. 867–881, 2012.

[22] C. Sirisomboonrat and K. Sinapiromsaran, "Breast cancer diagnosis using multi-attributed lens recursive partitioning algorithm," in *ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2012 10th International Conference on*, pp. 40–45, IEEE, 2012.

[23] J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on natural neighbor," *Knowledge-Based Systems*, vol. 92, pp. 71–77, 2016.

BIOGRAPHY

Name	Miss Warunya Kiangia
Date of Birth	12 June 1991
Place of Birth	Trang, Thailand
Education	Bachelor of Science (Mathematics), Chulalongkorn University, 2013
Publication	

- W. Kiangia, A. Luangsodsai, and K. Sinapiromsaran, “Weighted minimum consecutive pair of the extreme pole outlier factor,” in *Computer Science and Engineering Conference (ICSEC), 2016 International*, pp. 1–6, IEEE, 2016.