การตรวจหากลุ่มผิดปกติโดยใช้ระยะทางเพื่อนบ้านใกล้สุด

นายกายสิทธิ์ สิงห์กาล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

ANOMALOUS ASSEMBLAGE DETECTION USING NEAREST NEIGHBOR DISTANCE

Mr. Kayyasit Singkarn

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 2017

Thesis Title	ANOMALOUS ASSEMBLAGE DETECTION USING NEAREST NEIGHBOR DISTANCE
By	Mr. Kayyasit Singkarn
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
(Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

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

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

.... Examiner
(Arthorn Luangsodsai, Ph.D.)

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

กายสิทธิ์ สิงห์กาล : การตรวจหากลุ่มผิดปกติโดยใช้ระยะทางเพื่อนบ้านใกล้สุด. (ANOMALOUS ASSEMBLAGE DETECTION USING NEAREST NEIGHBOR DISTANCE) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. กรุง สินอภิรมย์สราญ, 81 หน้า.

ความผิดปกติของข้อมูลในงานวิจัยนี้ถูกนิยามด้วยระยะทางระหว่างข้อมูลสองตัว สำหรับบางเซตข้อมูล ข้อมูลผิดปกติอาจไม่แยกแบบโดดเดี่ยวและก่อตัวเป็นกลุ่มเล็ก ๆ กลุ่มผิดธรรมดา-ซี คือ กลุ่มของข้อมูลผิดปกติซึ่งสัมพันธ์กันโดยมีจำนวนข้อมูลในกลุ่มน้อยกว่าหรือเท่ากับซีเปอร์เซ็นของจำนวนข้อมูลทั้งหมด วิทยานิพนธ์นี้นำเสนอขั้นตอนวิธีการตรวจหากลุ่มผิดธรรมดาเรียกว่า ซีเอ็นดี โดยใช้ระยะห่างเพื่อนบ้านที่ใกล้ที่สุดแทนคะแนนความผิดปกติ ขั้นตอนวิธีนี้คำนวณดัชนีเคให้มีค่าเท่ากับฟังก์ชันพื้นของซีเปอร์เซ็นต์คูณจำนวนข้อมูลทั้งหมด และใช้ระยะทางเพื่อนบ้านใกล้สุดเคเพื่อแทนคะแนนของข้อมูลผิดปกติ หลังจากนั้น การปรับกราฟบ๊อกด้วยเมดคลับเปิลสำหรับการกระจายแบบเบ้ถูกใช้ในการคำนวณขีดแบ่งสำหรับการจับจุดผิดปกติ ประสิทธิภาพของซีเอ็นดีได้ถูกทดสอบกับชุดข้อมูลสองแบบ คือ เซตข้อมูลที่สังเคราะห์และเซตข้อมูลจริงจากเว็บไซต์ยูซีไอ เปรียบเทียบกับ ดับเบิ้ลยูโอเอฟ และ แอลโอเอฟ ผลการทดลองแสดงให้เห็นว่าประสิทธิภาพของ ซีเอ็นดี ดีกว่า ดับเบิ้ลยูโอเอฟ และ แอลโอเอฟ ภายใต้ความแม่นยำ การเรียกคืน และตัววัดเอฟหนึ่ง

ภาควิชา	คณิตศาสตร์และ	ลายมือชื่อนิสิต ....
	วิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
สาขาวิชา	คณิตศาสตร์ประยุกต์	
	และวิทยาการคณนา	
ปีการศึกษา	2560	

5871908723 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE, KEYWORDS : OUTLIER / ANOMALOUS ASSEMBLAGE DETECTION / K-NEAREST NEIGHBOR DISTANCE / MEDCOUPLE

KAYYASIT SINGKARN : ANOMALOUS ASSEMBLAGE DETECTION USING NEAREST NEIGHBOR DISTANCE. ADVISOR : ASSISTANT PROFESSOR KRUNG SINAPIROMSARAN, PH.D., 81 pp.

The outlierness of an instance in this thesis is defined based on the distance between two instances. For some datasets, outliers may not be isolated and formed small clusters. C-anomalous assemblage is a group of associated outliers having the number of instances less than or equal to C percent of the total instances. This thesis presents the anomalous assemblage detection algorithm called CND using a nearest neighbor distance for an anomalous score. The algorithm computes the index k equal to floor function of C percent times the total number of instances and uses the k^{th}-nearest neighbor distance for representing an anomalous score. Then, the adjusted boxplot based on medcouple for skew distribution is used to generate the threshold for detecting outliers. The performance of CND is tested on two types of datasets which are synthetic and real-world datasets from UCI website comparing with WOF and LOF. The experimental results show that CND is better than WOF and LOF on datasets based on precision, recall, and F_1-measure.

Department : Mathematics and Computer Science

Student's Signature ....

Advisor's Signature ....

Field of Study : Applied Mathematics and Computational Science

Academic Year : 2017

ACKNOWLEDGEMENTS

Firstly, I offer my sincere appreciation and special thanks to my thesis advisor, Assistant Professor Krung Sinapiromsaran, Ph.D.. This thesis has been successfully completed by his guidance.

Secondly, I am very thankful my thesis committees, Assistant Professor Boonyarit Intiyot, Ph.D., Arthorn Luangsodsai, Ph.D., and Assistant Professor Chumphol Bunkhumpornpat, Ph.D. for their useful comments and suggestions on my thesis.

Thirdly, I would like to thank Applied Mathematics and Computational Science (AMCS), Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University for financial support and research facilities.

Finally, I most grateful to my parents and my friends in the AMCS laboratory for all of their support throughout the period of this work.

CONTENTS

Page

CHAPTER

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Research motivation

“An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism” is a definition for an outlier which is stated by Hawkins ([1], [2]) in 1980. Outliers are also called as abnormalities, discordants, deviants, anomalies, exceptions, aberrations, surprises, peculiarities, or contaminants. The recognition of outliers provides useful information for data analysts because the outliers often disturb the accuracy of a predictive model, or causes inaccurate parameter estimation. An outlier detection (also known as an anomaly detection [3], [4], [5], [6]) is an important topic in data mining that identifies the outliers in a dataset. It appears in many real-world problems such as

- **Intrusion detection systems:** In computer systems, different kinds of data are collected such as network traffic which may contain small number of unusual behavior forming malicious activities. The detection of this activity is referred to as an intrusion detection.

- **Medical diagnosis:** In medical, the data is collected from many equipments such as MRI scans, PET scans, and ECG from patients. Unusual patterns of data may reflect a specific disease condition which defines as an outlier.

- **Earth science:** Spatial data is collected by satellites to track weather patterns, climate changes, or land cover patterns. The data may provide significant insights about unnatural environmental trends which are considered as outliers.

There are two types of outputs for an anomaly detection. The first type generates a score for an instance that represents the outlierness while the second type characterizes each instance as an outlier or a normal instance.

Anomalous score: This thesis will concentrate on the algorithm of the first type for an anomaly detection which is called a scoring algorithm. A score represents a level of "outlierness" of an instance which can be used as an outlierness ranking. This output does not define whether an instance is an outlier or a normal instance where a user must suggest a threshold to make a decision.

Numerous scoring algorithms were proposed where effective algorithms will provide an anomalous score of each instance according to outlierness of an instance. Well-known scoring algorithms are reviewed next.

Let $D \subseteq \mathbb{R}^n$ be a dataset having m instances where $p^{(i)}$ is the i^{th} instance in D for $i \in \{1, 2, ...m\}$.

1. Local-outlier-factor (LOF) [7] is proposed by Markus M. Breunig et al. in 2000 which is a popularly cited algorithm. The concept of LOF is based on the comparison of k-neighborhood density between an instance and its neighborhood. The author defined the local reachability density to represent a neighborhood density of each instance under the k-nearest neighbors. An outlier will have a lower density while a normal instance will have a higher density. An anomalous score of each instance is computed by the average summation of each k-nearest neighbor density divided by its density. The experimental results in their paper showed that a score of any outlier deviates so much from 1 whereas a score of any normal is close to 1. The disadvantage of LOF is specifying the suitable parameter k which represents the number of appropriate nearest neighbors. The crucial definitions for computing the LOF anomalous score of each instance in a dataset are shown next.

Note. Time complexity of LOF algorithm to compute all scores is $O(n^2)$.

The local reachability density of each instance is defined by

$$\mathrm{lrd}_k(p^{(i)}) = 1 \ / \left(\frac{\sum_{p^{(j)} \in N_k(p^{(i)})} \text{reach-dist}_k(p^{(i)}, p^{(j)})}{\left|N_k(p^{(i)})\right|} \right)$$

where reach-dist$_k(p^{(i)}, p^{(j)}) = \max\{\text{k-distance}(p^{(j)}), d(p^{(i)}, p^{(j)})\}$ such that k-distance$(p^{(i)})$ is distance between $p^{(i)}$ and its k^{th}-nearest neighbor, and $N_k(p^{(i)}) = \{p^{(j)} \in D \backslash p^{(i)} \mid d(p^{(i)}, p^{(j)}) \leq k\text{-distance}(p^{(i)})\}$.

The local outlier factor (LOF anomalous score) of each instance is defined by

$$\mathrm{LOF}_k(p^{(i)}) = \frac{\sum_{p^{(j)} \in N_k(p^{(i)})} \frac{\mathrm{lrd}_k(p^{(j)})}{\mathrm{lrd}_k(p^{(i)})}}{\left|N_k(p^{(i)})\right|}.$$

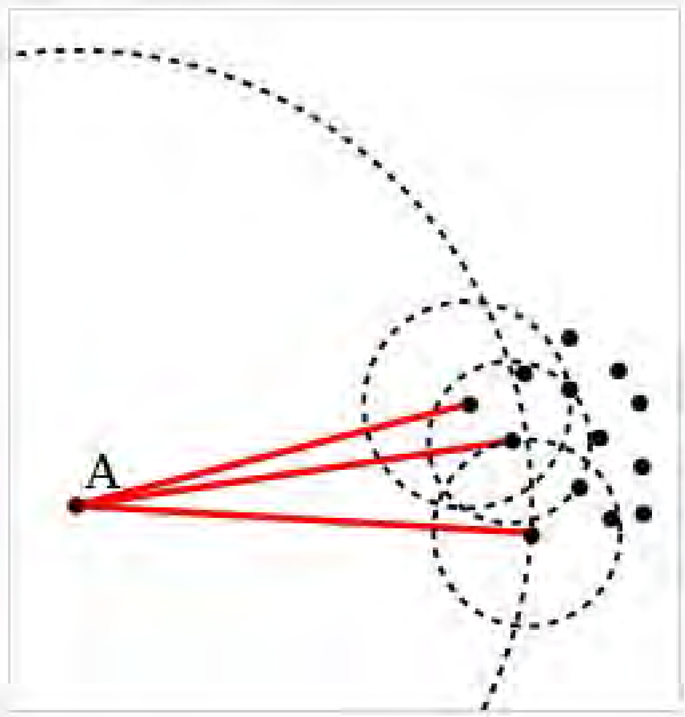


Figure 1.1: Basic idea of LOF is the comparison between the local density of an instance and its neighborhoods. Instance A has a low density than its neighborhood based on $k = 3$, it will be considered as an outlier.
source: www.en.wikipedia.org/wiki/Local-outlier-factor

2. Histogram-based outlier score (HBOS) [8] is proposed by Markus Goldstein and Andreas Dengel in 2012. It is a linear-time algorithm to compute the anomalous scores. The concept of HBOS based on the conversion of instances to the histogram on each axis. An anomalous score of each instance is computed from the height of the histogram which it is located. The disadvantage of HBOS is the setting of parameter k representing the number of histogram bins.

Note. Time complexity of HBOS algorithm to compute the scores is $O(n)$.

HBOS anomalous score of each instance $p^{(i)}$ is defined by

$$\text{HBOS}(p^{(i)}) = \sum_{t=1}^{d} \log\left(\frac{1}{\text{hist}_t(p^{(i)})}\right)$$

where d is the number of attributes and $\text{hist}_t(p^{(i)})$ is the height of histogram which the instance $p^{(i)}$ is located along the t^{th} attribute. Note that the logarithm is used for reducing the errors from floating point precision causing when the score is very high.

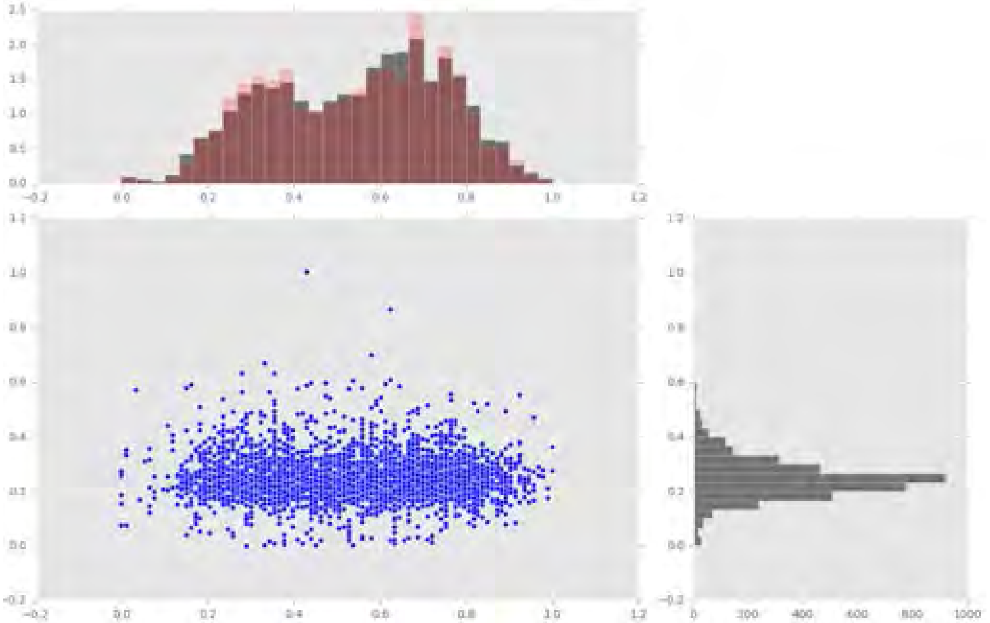


Figure 1.2**:** The histogram along each attribute on a dataset in $\mathbb{R}^2$.
source: www.shahramabyari.com/detecting-outliers-in-high-dimensional-data-sets

3. Ordered distance difference outlier factor (OOF) [9] is proposed by Nattorn Buthong et al. in 2013. The concept of OOF based on the ordered distance of an instance along the other instances. The author defined the ordered distance matrix which is the k-nearest neighbor distance matrix of each instance and constructed the difference of the ordered distance matrix to compute OOF anomalous score. An anomalous score is computed from the average of difference of ordered distance with respect to every instance in a dataset. OOF does not require a parameter. The important definitions to generate OOF anomalous score of each instance in a dataset are shown next.

Note. Time complexity of OOF algorithm to compute the scores is $O(n^2 \log n)$.

The ordered distance matrix of a dataset D is defined as

$$\text{OrderedMtx}(D) = \begin{bmatrix} 0 & d_{1,j_2^{(1)}} & d_{1,j_3^{(1)}} & \cdots & d_{1,j_m^{(1)}} \\ 0 & d_{2,j_2^{(2)}} & d_{2,j_3^{(2)}} & \cdots & d_{2,j_m^{(2)}} \\ 0 & d_{3,j_2^{(3)}} & d_{3,j_3^{(3)}} & \cdots & d_{3,j_m^{(3)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & d_{m,j_2^{(m)}} & d_{m,j_3^{(m)}} & \cdots & d_{m,j_m^{(m)}} \end{bmatrix}$$

where $d_{i,j} = d(p^{(i)}, p^{(j)})$ such that $0 = d_{i,j_1^{(i)}} \leq d_{i,j_2^{(i)}} \leq d_{i,j_3^{(i)}} \leq \cdots \leq d_{i,j_m^{(i)}}$ for $i \in \{1, 2, ..., m\}$.

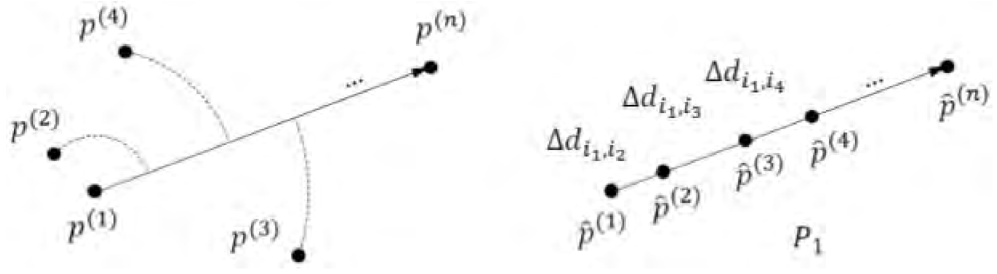


Figure 1.3: The difference of ordered distance [9].

The difference of the ordered distance matrix is defined as

$$\text{DiffOrderedMtx}(D) = \begin{bmatrix} 0 & \Delta d_{1,j_2^{(1)}} & \Delta d_{1,j_3^{(1)}} & \cdots & \Delta d_{1,j_m^{(1)}} \\ 0 & \Delta d_{2,j_2^{(2)}} & \Delta d_{2,j_3^{(2)}} & \cdots & \Delta d_{2,j_m^{(2)}} \\ 0 & \Delta d_{3,j_2^{(3)}} & \Delta d_{3,j_3^{(3)}} & \cdots & \Delta d_{3,j_m^{(3)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \Delta d_{m,j_2^{(n)}} & \Delta d_{m,j_3^{(m)}} & \cdots & \Delta d_{m,j_m^{(m)}} \end{bmatrix}$$

where $\Delta d_{i,j_k^{(i)}} = d_{i,j_k^{(i)}} - d_{i,j_{k-1}^{(i)}}$ for $k \in \{2, 3, ..., m\}$.

OOF anomalous score of each instance p is computed by

$$\frac{\sum_{i=1}^{m} \Delta d_{i,index(p)}(i)}{m-1}$$

It is the average of the difference of ordered distance.

4. Weighted minimum consecutive pair of the extreme pole outlier factor (WOF) [10] is proposed by Warunya Kiangia et al. in 2016. The concept of WOF based on the projection of all instances to the vector core. The author defined the vector core which is a vector of two farthest instances (extreme poles). An anomalous score of each instance is computed from the minimum weighted of its along each side of the projection on the vector core. WOF also does not require a parameter. The important definitions for computing WOF anomalous score of each instance in a dataset are shown next.

Note. Time complexity of WOF to compute the scores is equal to $O(n^2)$.

The extreme pole: Let $e_1 \in \{1, 2, 3, ..., m\}$ and $e_2 \in \{1, 2, 3, ..., m\}$. If $d(p^{(e_1)}, p^{(e_2)}) = \max\{d(p^{(i)}, p^{(j)})\}$, then $p^{(e_1)}$ and $p^{(e_2)}$ are extreme poles. In addition, the vector core is a vector that starts from one extreme pole to another extreme pole.

The projection of instances to the vector core is defined as

$$\mathrm{OrdList}(D, e) = [d(p^{(e)}, p^{(i_1)}), d(p^{(e)}, p^{(i_2)}), ..., d(p^{(e)}, p^{(i_m)})].$$

where $e \in \{e_1, e_2\}$ is an index of the extreme pole and $i_1, i_2, ..., i_m \in \{1, 2, ..., m\}$. WLOG, $i_1 = 1, i_2 = 2, ..., i_m = m$ such that $0 = d(p^{(e)}, p^{(e)}) \leq d(p^{(e)}, p^{(i_1)}) \leq d(p^{(e)}, p^{(i_2)}) \leq ... \leq d(p^{(e)}, p^{(i_{m-1})}) \leq d(p^{(e)}, p^{(i_m)})$. See the example in Figure 1.4 - 1.5.

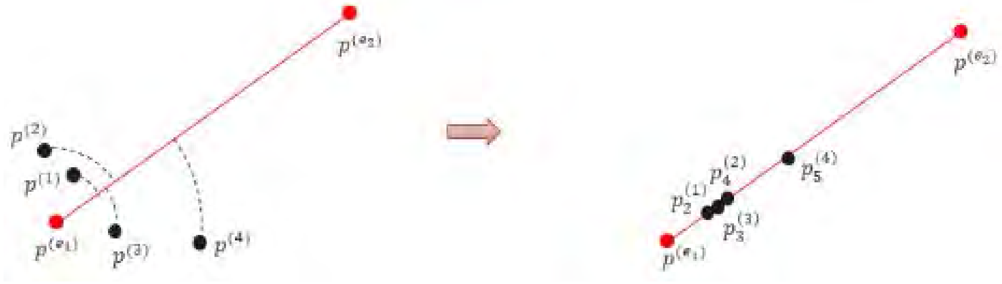


Figure 1.4**:** The projection with respect to $p^{(e_1)}$ [10].

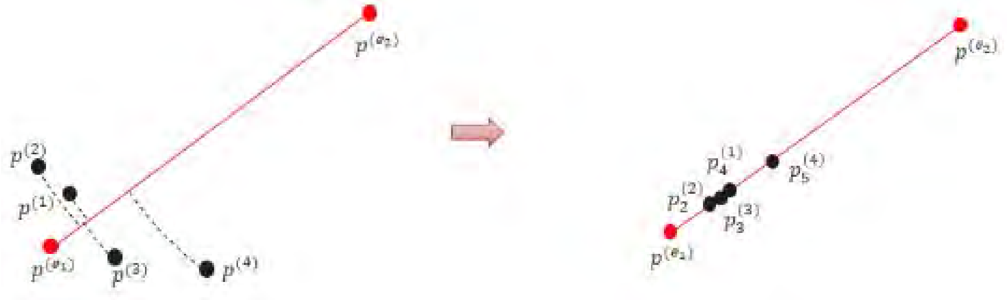


Figure 1.5**:** The projection with respect to $p^{(e_2)}$ [10].

The projected order score on the vector core from the extreme pole is defined as

$$\mathrm{OF}_e(p^{(k)}) = \begin{cases} d(p^{(e)}, p^{(i_2)}) - d(p^{(e)}, p^{(i_1)}) & \text{if } k = i_1 \\ \frac{(d(p^{(e)}, p^{(i_k)}) - d(p^{(e)}, p^{(i_{k-1})}))(i_k - 1)}{m-1} + \frac{(d(p^{(e)}, p^{(i_{k+1})}) - d(p^{(e)}, p^{(i_k)}))(m - i_k)}{m-1} & \text{if } k \in \{i_2, ..., i_{m-1}\} \\ d(p^{(e)}, p^{(i_m)}) - d(p^{(e)}, p^{(i_{m-1})}) & \text{if } k = i_m. \end{cases}$$

WOF anomalous score of each instance is computed from

$$\mathrm{WOF}(p^{(k)}) = \frac{\mathrm{OF}_{e_1}(p^{(k)}) + \mathrm{OF}_{e_2}(p^{(k)})}{2}.$$

Binary labels: The second type of output is a binary label identifying whether an instance is an outlier or a normal instance. Typically, the first type of anomalous algorithm could transform into the second type using a simple criterion for labeling an instance which is computed from the cutoff or the threshold of the anomalous scores.

After the anomalous scores of instances are computed from a scoring algorithm, the criteria for labeling the class of each instance can be applied as follows.

5. A box plot or boxplot is a popular tool (which proposed by Tukey, J.W. [11] in 1977) to virtualize the distribution of a continuous variable based on the interquartile range. It displays the distribution of data using the five descriptive statistics which are the minimum, the first quartile (Q_1), the median or second quartile (Q_2), the third quartile (Q_3), and the maximum. The boxplot is constructed by drawing the center rectangle as the box covering the first quartile to the third quartile where the length of this box is equal to the interquartile range (IQR) ($Q_3 - Q_1$), a segment inside the interquartile range shows the median (Q_2), the below of box shows the location of the minimum, and the above of box shows the location of the maximum. See Figure 1.6.

The criteria for identifying normal instances based on the boxplot is presented by the interval

$$[Q_1 - 1.5IQR, Q_3 + 1.5IQR].$$

The scores outside this interval will be considered as the outliers. This threshold works well with the normal distribution. See Figure 1.7.

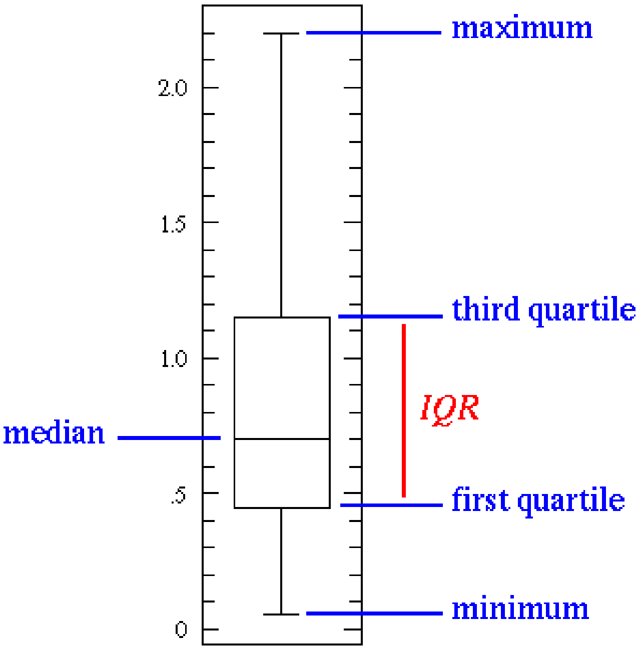


Figure 1.6**:** Components of a boxplot.
source: www.physics.csbsju.edu/stats/box2.html

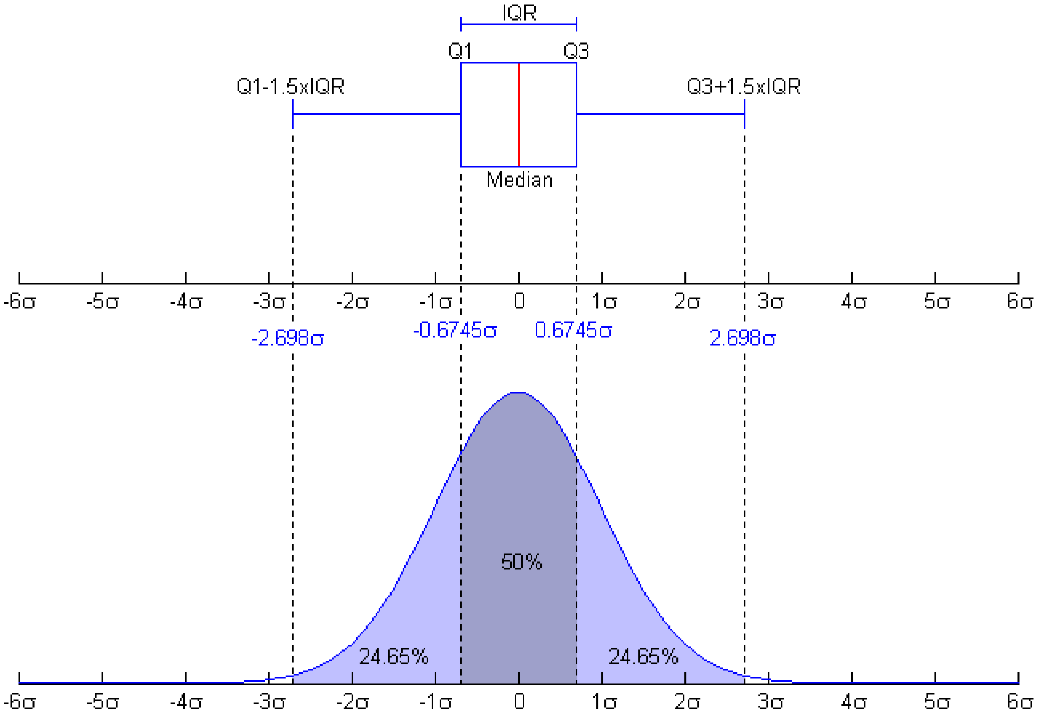


Figure 1.7**:** The boxplot for the normal distribution.
source: www.en.wikipedia.org/wiki/Box_plot

6. An adjustment of the boxplot (also called the adjusted boxplot [12]) is presented by Hubert et al. in 2008 which is the generalized criteria for detecting the outliers for various distributions. It uses the medcouple (MC) to measure the skewness of the distribution. The experimental results shown that the model from the exponential formulation is the best model.

$[Q_1 - 1.5e^{-4MC}IQR; Q_3 + 1.5e^{3MC}IQR]$ for $MC \geq 0$, and
$[Q_1 - 1.5e^{-3MC}IQR; Q_3 + 1.5e^{4MC}IQR]$ for $MC < 0$.

An instance having the score outside the interval will be labeled as the outlier.

Our work

The outlier may be classified as a single outlier or associated as a group of outliers. In this thesis, a group of associated outliers is defined as an anomalous assemblage. The number of outliers in an anomalous assemblage is very small and far away comparing with other clusters in a dataset. Moreover, C-anomalous assemblage is defined as the anomalous assemblage having the number of instances less than or equal to C percent of the number of instances in a dataset. Note that a dataset may have multiple anomalous assemblages.

Most of the above scoring algorithms are not designed to effectively detect the anomalous assemblages. Only LOF with the appropriate parameter k may determine the proper scores of these outliers. Consequently, this thesis proposes a new anomaly detection algorithm called CND which is designed to effectively detect a single outlier and a group of outliers. The basic idea is based on the k-nearest neighbor distance which is used to represent an anomalous score of each instance in a dataset where k is set to be equal to the floor function of the C percent of the total number of instances. For labeling the class of each instance as an outlier or a normal instance, the upper threshold from the adjusted boxplot [12] based on the medcouple for skew distribution is used.

1.2 Research objective

The goal of this thesis is to propose a new anomaly detection algorithm for detecting the point outliers on a finite continuous-valued attribute dataset in $\mathbb{R}^n$ using the distance-based approach. The C-anomalous assemblage detection algorithm called CND is proposed where it is designed for effectively detecting the anomalous assemblages. Moreover, the performance of the proposed algorithm based on time complexity and the capability for detecting the outliers are evaluated and compared with other algorithms.

1.3 Thesis overview

This thesis is divided into five chapters. Chapter I presents the introduction. Chapter II shows the background knowledge. Chapter III describes the definitions, basic idea, and the proposed algorithm. Chapter IV shows the experimental results. The last chapter provides the conclusion and future work.

CHAPTER II

BACKGROUND KNOWLEDGE

This chapter covers background knowledge which includes the Minkowski distance, k-nearest neighbor, outlier, detection threshold, and performance measurements.

2.1 Minkowski distance

Let $D \subseteq \mathbb{R}^n$ be a finite dataset with m instances and $p^{(i)} = (p_1^{(i)}, p_2^{(i)}, ..., p_n^{(i)})$ be i^{th} instance in D where $i \in \{1, 2, ..., m\}$.

Definition 2.1. (Minkowski distance ([13], [14]))
The Minkowski distance of order q between two instances $p^{(i)}$ and $p^{(j)}$ is defined as

$$d_q(p^{(i)}, p^{(j)}) = \sqrt[q]{\sum_{t=1}^{n} \left| p_t^{(i)} - p_t^{(j)} \right|^q}$$

It is often used with order q equals 1 or 2. The Minkowski distance of order $q = 1$ is called the Manhattan distance and the Minkowski distance of order $q = 2$ is called the Euclidean distance.

$$d_1(p^{(i)}, p^{(j)}) = \sum_{t=1}^{n} \left| p_t^{(i)} - p_t^{(j)} \right| \quad \text{and} \quad d_2(p^{(i)}, p^{(j)}) = \sqrt{\sum_{t=1}^{n} (p_t^{(i)} - p_t^{(j)})^2}.$$

Example 2.1. Let $p^{(1)} = (1,1)$ and $p^{(2)} = (2,2)$ are the instances in $\mathbb{R}^2$. Manhattan and Euclidean distance between $p^{(1)}$ and $p^{(2)}$ can be computed as following.

$$\begin{aligned}
\text{Manhattan distance:} \qquad d_1(p^{(1)}, p^{(2)}) &= \sum_{t=1}^{2} \left| p_t^{(1)} - p_t^{(2)} \right| \\
&= \left| p_1^{(1)} - p_1^{(2)} \right| + \left| p_2^{(1)} - p_2^{(2)} \right| \\
&= |1-2| + |1-2| \\
&= 2 \\
\text{Euclidean distance:} \qquad d_2(p^{(1)}, p^{(2)}) &= \sqrt{\sum_{t=1}^{2} (p_t^{(1)} - p_t^{(2)})^2} \\
&= \sqrt{(p_1^{(1)} - p_1^{(2)})^2 + (p_2^{(1)} - p_2^{(2)})^2} \\
&= \sqrt{(1-2)^2 + (1-2)^2} \\
&= \sqrt{1+1} \\
&= \sqrt{2} \\
&\approx 1.414
\end{aligned}$$

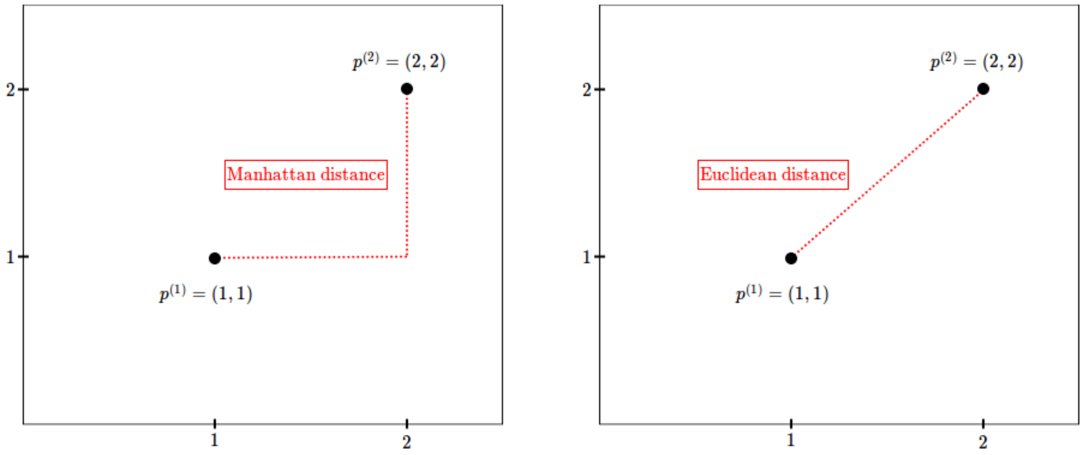


Figure 2.1: Demonstration of Manhattan distance and Euclidean distance in $\mathbb{R}^2$.

2.2 Nearest neighbor

The nearest neighbor ([15], [16]) is a popular idea used in many applications of data mining. The basic idea is based on the proximity between an instance and its neighborhood. Generally, the k-nearest neighbors of an instance are the k closest instances to that instance in a feature space. See examples in Figure 2.2.

Example 2.2.

- The 1-nearest neighbor of $p^{(1)}$ is $p^{(2)}$.
- The 2-nearest neighbors of $p^{(1)}$ are $p^{(2)}$ and $p^{(3)}$.
- The 3-nearest neighbors of $p^{(1)}$ are $p^{(2)}$, $p^{(3)}$, and $p^{(4)}$.

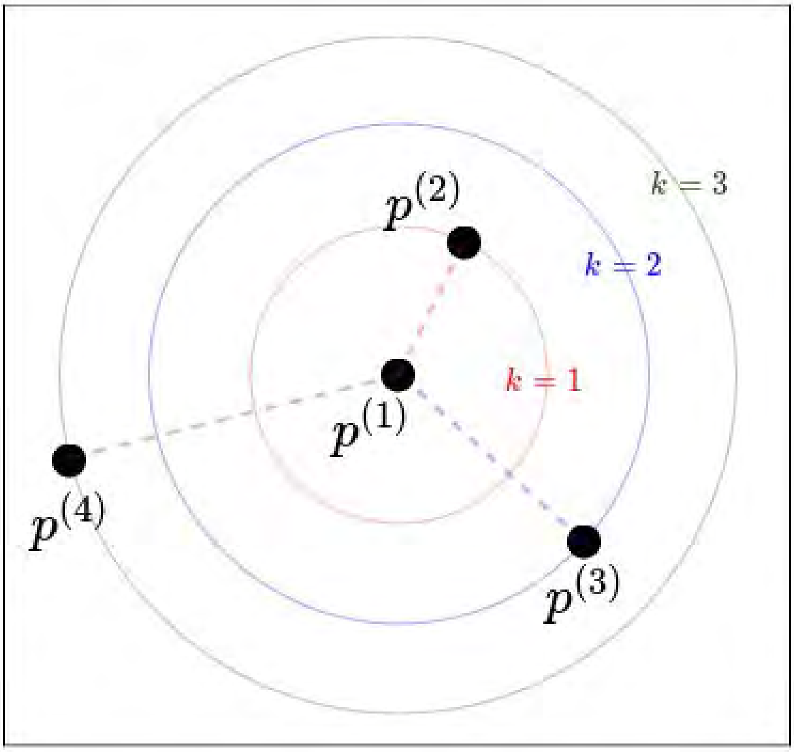


Figure 2.2: The k-nearest neighbors of $p^{(1)}$ for $k = 1, 2, 3$ based on Euclidean distance in $\mathbb{R}^2$.

2.3 Outlier

An outlier can be classified into three types as follows.

2.3.1 Point outlier

A point outlier is an individual instance which can be considered oddity with respect to the rest of instances. See examples in Figure 2.3.

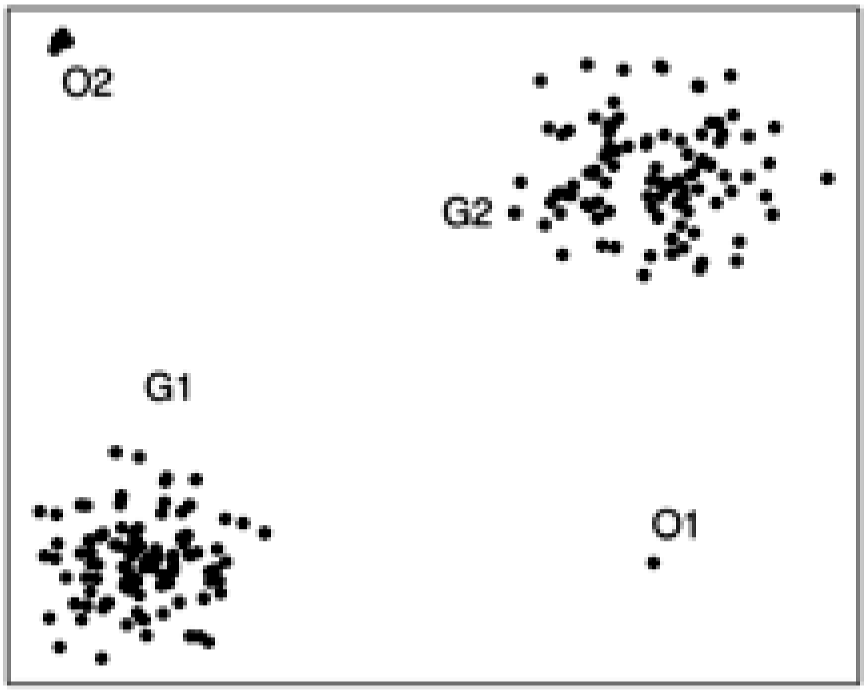


Figure 2.3: O_1 and O_2 are considered as the point outliers.
source: www.ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/I/m/Two-dimensional-Outliers-Example.png

2.3.2 Contextual outlier

A contextual outlier is an instance which can be considered as an outlier in a specific context where the sequence of instances in a dataset is important. See Figure 2.4.

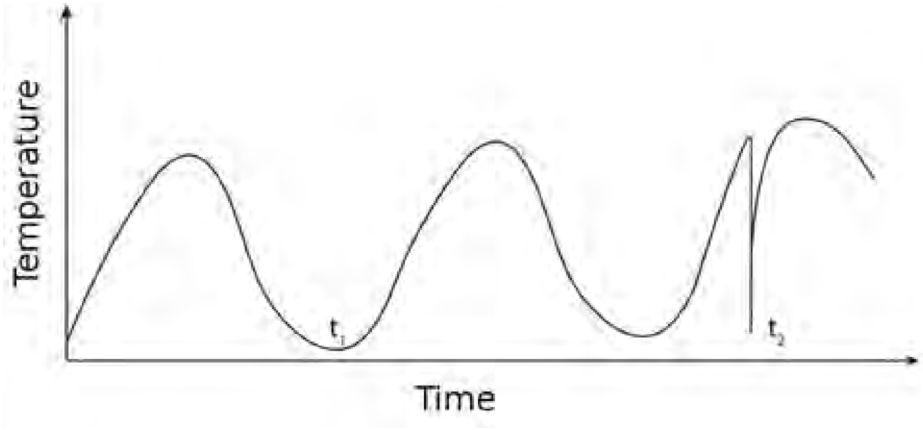


Figure 2.4**:** On time-series dataset of the temperatures, t_1 and t_2 are the same value but only t_2 will be considered as an outlier.
source: www.upload.wikimedia.org/wikipedia/commons/6/63/Contextual-Outlier.png

2.3.3 **Collective outlier**

A collective outlier is a collection of instances which can be considered as an outlier with respect to the entire dataset but each instance inside a collective outlier may not be an outlier by itself alone. An example of the collective outlier from the human electrocardiogram is shown in Figure 2.5.

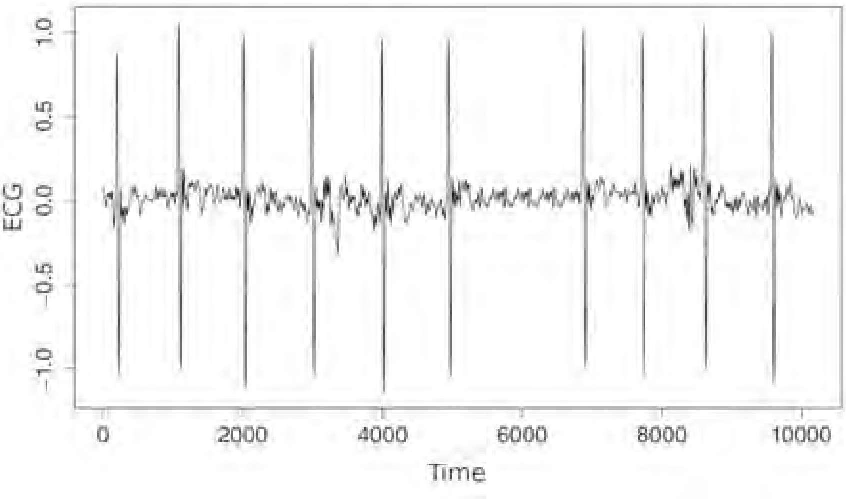


Figure 2.5**:** The values in the interval [5000, 7000] represent the collective outlier.
source: www.wikimedia.org/wikipedia/commons/thumb/4/4f/Collective-Outlier.png/ 1024px-Collective-Outlier.png

2.4 Anomaly detection

Anomaly detection (also called outlier detection) is the process of identifying outliers in a dataset. It can be categorized as supervised anomaly detection, semi-supervised anomaly detection and unsupervised anomaly detection.

2.4.1 Supervised anomaly detection

Supervised anomaly detection (is also called the classification technique) requires a training dataset model where each instance is labeled as a normal instance or an outlier. Each instance in a test dataset will be identified as a normal or an outlier based on the labeled in a training dataset.

2.4.2 Semi-supervised anomaly detection

Semi-supervised anomaly detection requires a training dataset model which contains only the normal instances. If an instance in a test dataset is similar to the instances in a training dataset, it will be considered as a normal instance. If an instance in a test dataset is different from instances in a training dataset, it will be considered as an outlier.

2.4.3 Unsupervised anomaly detection

Unsupervised anomaly detection detects the outliers in a test dataset under the assumption that the majority of instances in a dataset are the normal instances. It does not require the labeled instances from a training dataset. An instance which is not similar to the majority of instances will be considered as an outlier.

NOTE. A training dataset is a dataset used for training a model, while a test dataset contains instances from the same population however it normally contains no instances from a training dataset.

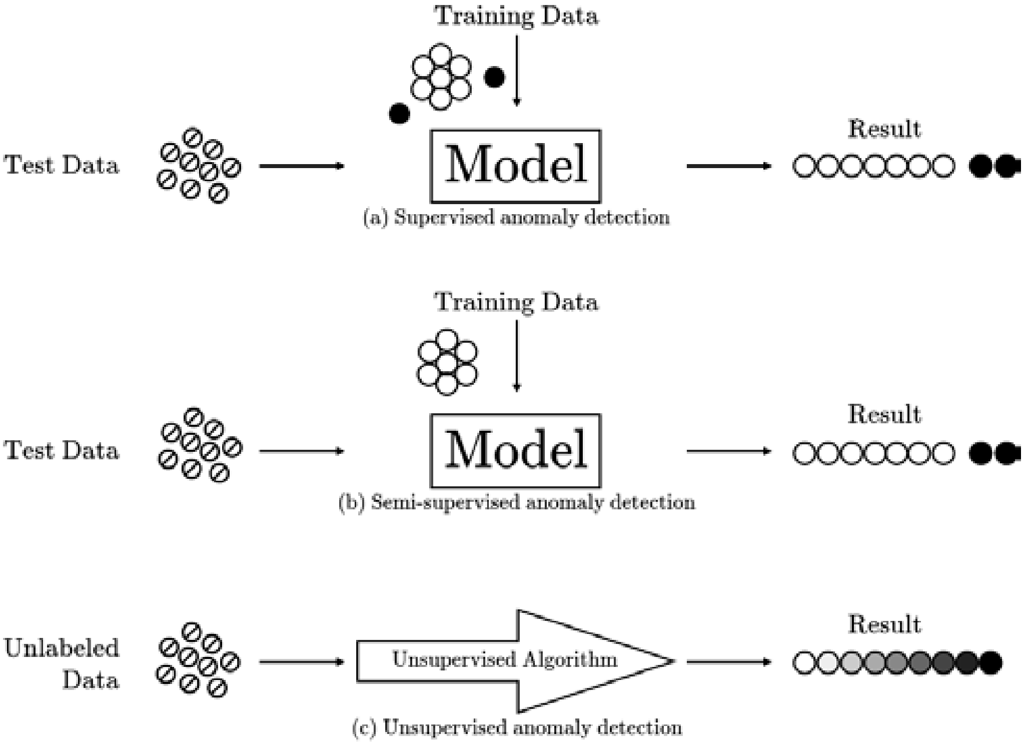


Figure 2.6: ⊘ is an unknown instance, the white points are the normal instances, and the black points are the outliers.

2.5 Anomalous score algorithm

Many techniques are introduced for computing anomalous scores of instances in a dataset in the research literatures. The popular algorithm is LOF and the latest algorithm is WOF in 2017 which are described next.

2.5.1 Local outlier factor (LOF)

The local outlier factor (LOF) [7] is a popular anomalous scoring algorithm which was proposed by Markus M. Breunig, et al. in 2000. The idea is based on the comparison between the local density of an instance and its neighborhood. An anomalous score of each instance is represented by the local-outlier-factor score (also called LOF score) which is computed from the ratio between the local density of this instance and its neighborhood. An outlier is an instance which has a lower local density. LOF needs a parameter k representing the number of nearest neighbors of an instance to run the algorithm. The definitions to generate the anomalous scores are shown next.

Definition 2.2. (k-distance)

For any positive integer k, the k-distance of an instance $p^{(i)}$ denoted as k-distance$(p^{(i)})$ is defined as the distance between $p^{(i)}$ and $p^{(j)}$ such that

(i) for at least k instances $p^{(j')} \in D\backslash p^{(i)}$, it holds that $d(p^{(i)}, p^{(j')}) \leq d(p^{(i)}, p^{(j)})$ and

(ii) for at most $k-1$ instances $p^{(j')} \in D\backslash p^{(i)}$, it holds that $d(p^{(i)}, p^{(j')}) < d(p^{(i)}, p^{(j)})$.

Definition 2.3. (k-distance neighborhood)

Given the k-distance of $p^{(i)}$, the k-distance neighborhood of $p^{(i)}$, denoted by $N_k(p^{(i)})$, contains every instance whose distance from $p^{(i)}$ is not greater than the k-distance of $p^{(i)}$.

$$N_k(p^{(i)}) = \{p^{(j)} \in D\backslash p^{(i)} \mid d(p^{(i)}, p^{(j)}) \leq k\text{-distance}(p^{(i)})\}.$$

An instance $p^{(j)} \in N_k(p^{(i)})$ is called a k-nearest neighbor of $p^{(i)}$.

Definition 2.4. (Reachability distance)

Let k be a positive integer. The reachability distance of $p^{(i)}$ with respect to $p^{(j)}$ (see the example in Figure 2.7) is defined as

$$\text{reach-dist}_k(p^{(i)}, p^{(j)}) = \max\{k\text{-distance}(p^{(j)}), d(p^{(i)}, p^{(j)})\}.$$

Definition 2.5. (Local reachability density)

The local reachability density of $p^{(i)}$ is defined as

$$\text{lrd}_k(p^{(i)}) = 1 \ / \left(\frac{\sum\limits_{p^{(j)} \in N_k(p^{(i)})} \text{reach-dist}_k(p^{(i)}, p^{(j)})}{\left|N_k(p^{(i)})\right|} \right)$$

which is the inverse of the average reachability distance of $p^{(i)}$. It is the local density of an instance based on its k-nearest neighbors.

Definition 2.6. (Local outlier factor)

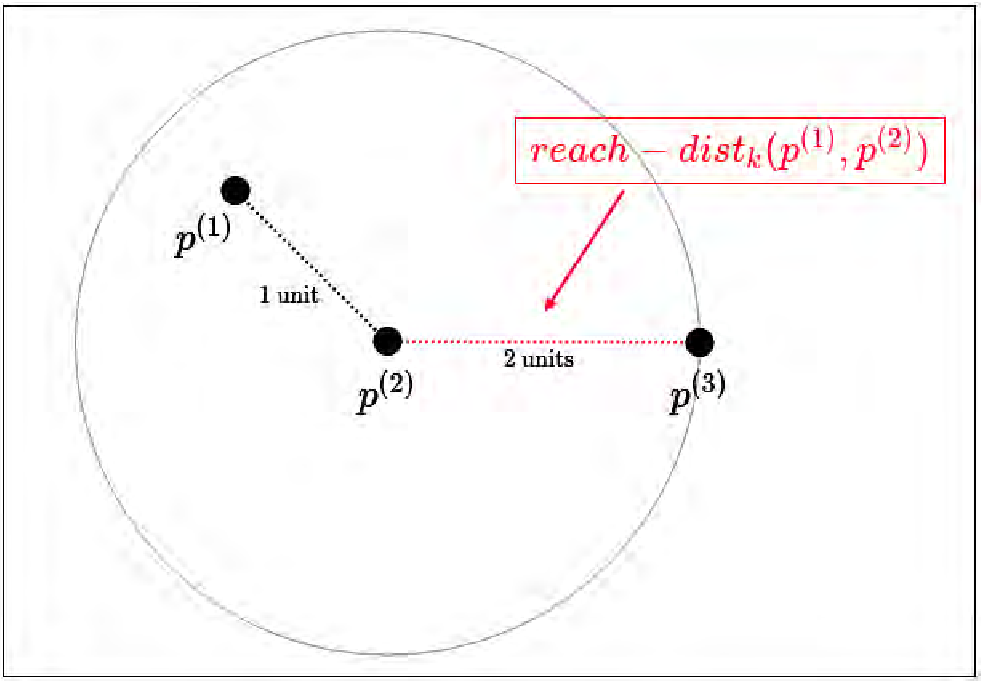


Figure 2.7**:** The reachability distance of $p^{(1)}$ with respect to $p^{(2)}$ for $k = 2$ with the Euclidean distance in $\mathbb{R}^2$.

The local outlier factor of $p^{(i)}$ is defined as

$$\mathrm{LOF}_k(p^{(i)}) = \frac{\displaystyle\sum_{p^{(j)} \in N_k(p^{(i)})} \frac{\mathrm{lrd}_k(p^{(j)})}{\mathrm{lrd}_k(p^{(i)})}}{\left|N_k(p^{(i)})\right|}.$$

It is the average of ratio of the local reachability density of $p^{(i)}$ to its k-nearest neighbors. The local outlier factor is used to represent LOF anomalous score of each instance in a dataset.

Example 2.3**.** Let $D \subseteq \mathbb{R}^2$ be a dataset having five instances which are $p^{(1)} = (-0.366, 0.046)$, $p^{(2)} = (3.598, 3.103), p^{(3)} = (3.242, 4.26), p^{(4)} = (4.7, 2.164)$ and $p^{(5)} = (4.414, 4.562)$. LOF anomalous scores of all instances in D using the parameter $k = 2$ based on the Euclidean distance can be computed as follows.

Step 1: Compute all distances between any two instances.

$d_2(p^{(1)}, p^{(2)}) = 5.006, d_2(p^{(1)}, p^{(3)}) = 5.548, d_2(p^{(1)}, p^{(4)}) = 5.491, d_2(p^{(1)}, p^{(5)}) = 6.576$

$d_2(p^{(2)}, p^{(3)}) = 1.211, d_2(p^{(2)}, p^{(4)}) = 1.448, d_2(p^{(2)}, p^{(5)}) = 1.672$

$d_2(p^{(3)}, p^{(4)}) = 2.553, d_2(p^{(3)}, p^{(5)}) = 1.211$

$d_2(p^{(4)}, p^{(5)}) = 2.415$

Step 2: Compute the 2-distance of each instance.

2-distance($p^{(1)}$) $= 5.491$,

2-distance($p^{(2)}$) $= 1.448$,

2-distance($p^{(3)}$) $= 1.211$,

2-distance($p^{(4)}$) $= 2.415$,

2-distance($p^{(5)}$) $= 1.672$

Step 3: Construct the 2-distance neighborhood of each instance.

$N_2(p^{(1)}) = \{p^{(2)}, p^{(4)}\}$,

$N_2(p^{(2)}) = \{p^{(3)}, p^{(4)}\}$,

$N_2(p^{(3)}) = \{p^{(2)}, p^{(5)}\}$,

$N_2(p^{(4)}) = \{p^{(2)}, p^{(5)}\}$,

$N_2(p^{(5)}) = \{p^{(3)}, p^{(2)}\}$

Step 4: Compute the reachability distance of each instance w.r.t. other instances.

$$\begin{aligned}
\text{reach-dist}_2(p^{(1)}, p^{(1)}) &= \max\{\text{2-distance}(p^{(1)}), d_2(p^{(1)}, p^{(1)})\}\\
&= \max\{5.491, 0\} = 5.491\\
\text{reach-dist}_2(p^{(1)}, p^{(2)}) &= \max\{\text{2-distance}(p^{(2)}), d_2(p^{(1)}, p^{(2)})\}\\
&= \max\{1.448, 5.006\} = 5.006\\
\text{reach-dist}_2(p^{(1)}, p^{(3)}) &= \max\{\text{2-distance}(p^{(3)}), d_2(p^{(1)}, p^{(3)})\}\\
&= \max\{1.211, 5.548\} = 5.548\\
\text{reach-dist}_2(p^{(1)}, p^{(4)}) &= \max\{\text{2-distance}(p^{(4)}), d_2(p^{(1)}, p^{(4)})\}\\
&= \max\{2.415, 5.491\} = 5.491\\
\text{reach-dist}_2(p^{(1)}, p^{(5)}) &= \max\{\text{2-distance}(p^{(5)}), d_2(p^{(1)}, p^{(5)})\}\\
&= \max\{1.672, 6.576\} = 6.576.
\end{aligned}$$

Similarly perform for reach-dist$_2(p^{(2)}, p^{(j)})$, reach-dist$_2(p^{(3)}, p^{(j)})$, reach-dist$_2(p^{(4)}, p^{(j)})$, and reach-dist$_2(p^{(5)}, p^{(j)})$. Then, we will get

reach-dist$_2(p^{(2)}, p^{(j)}) = 5.491, 1.448, 1.211, 2.415, 1.672$ for $j = 1, 2, 3, 4, 5$ respectively.

reach-dist$_2(p^{(3)}, p^{(j)}) = 5.548, 1.448, 1.211, 2.553, 1.672$ for $j = 1, 2, 3, 4, 5$ respectively.

reach-dist$_2(p^{(4)}, p^{(j)}) = 5.491, 1.448, 2.553, 2.415, 2.415$ for $j = 1, 2, 3, 4, 5$ respectively.

reach-dist$_2(p^{(5)}, p^{(j)}) = 6.576, 1.672, 1.211, 2.415, 1.672$ for $j = 1, 2, 3, 4, 5$ respectively.

Step 5: Compute the local reachability density of each instance.

$$\begin{aligned}\text{lrd}_2(p^{(1)}) &= 1 \;/\; \left(\frac{\displaystyle\sum_{p^{(j)} \in N_2(p^{(i)})} \text{reach-dist}_2(p^{(1)}, p^{(j)})}{\left|N_2(p^{(1)})\right|} \right) \\ &= 1 \;/\; \frac{\text{reach-dist}_2(p^{(1)}, p^{(2)}) + \text{reach-dist}_2(p^{(1)}, p^{(4)})}{2} \\ &= 1 \;/\; \frac{5.006 + 5.491}{2} \\ &= 0.191.\end{aligned}$$

Similarly compute for lrd$_2(p^{(2)})$, lrd$_2(p^{(3)})$, lrd$_2(p^{(4)})$, and lrd$_2(p^{(5)})$. Then lrd$_2(p^{(2)}) = 0.552$, lrd$_2(p^{(3)}) = 0.641$, lrd$_2(p^{(4)}) = 0.518$, and lrd$_2(p^{(5)}) = 0.694$.

Step 6: Compute the local outlier factor of each instance.

$$\begin{aligned}\text{LOF}_2(p^{(1)}) &= \frac{\displaystyle\sum_{p^{(j)} \in N_2(p^{(1)})} \frac{\text{lrd}_2(p^{(j)})}{\text{lrd}_2(p^{(1)})}}{\left|N_2(p^{(1)})\right|}. \\ &= \frac{\frac{\text{lrd}_2(p^{(2)})}{\text{lrd}_2(p^{(1)})} + \frac{\text{lrd}_2(p^{(4)})}{\text{lrd}_2(p^{(1)})}}{2} \\ &= \frac{\frac{0.552}{0.191} + \frac{0.518}{0.191}}{2} \\ &= 2.801\end{aligned}$$

Similarly compute for LOF$_2(p^{(2)})$, LOF$_2(p^{(3)})$, LOF$_2(p^{(4)})$, and LOF$_2(p^{(5)})$. Then LOF$_2(p^{(2)}) = 1.05$, LOF$_2(p^{(3)}) = 0.971$, LOF$_2(p^{(4)}) = 1.203$, and LOF$_2(p^{(5)}) = 0.859$.

2.5.2 Weighted minimum consecutive pair of the extreme poles outlier factor (WOF)

The weighted minimum consecutive pair of the extreme pole outlier factor (WOF) is proposed by Kiangia et al. [10] in 2016 which is a parameter-free algorithm. The basic idea is based on the projection of all instances to the vector core. It computes an anomalous score from the weighted minimum consecutive pair on each side along the projection of instances on the vector core.

Definition 2.7. (The matrix of distance)
The matrix of distance of a dataset D with m instances is defined by

$$M = [d_{ij}]_{m \times m}$$

where $d_{ij} = d(p^{(i)}, p^{(j)})$ for $p^{(i)}, p^{(j)} \in D$ and $i, j \in \{1, 2, 3, ..., m\}$.

The matrix of distance can be rewritten as

$$M = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1m} \\ d_{21} & 0 & d_{23} & \cdots & d_{2m} \\ d_{31} & d_{32} & 0 & \cdots & d_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & 0 \end{bmatrix}.$$

Definition 2.8. (The extreme pole)
Given $e_1 \in \{1, 2, 3, ..., m\}$ and $e_2 \in \{1, 2, 3, ..., m\}$ such that $d(p^{(e_1)}, p^{(e_2)}) = \max\{d(p^{(i)}, p^{(j)})\}$, $i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., m\}$, then $p^{(e_1)}$ and $p^{(e_2)}$ are extreme poles.

Definition 2.9. (The vector core)
The vector core is a vector that starts from one extreme pole to another extreme pole.

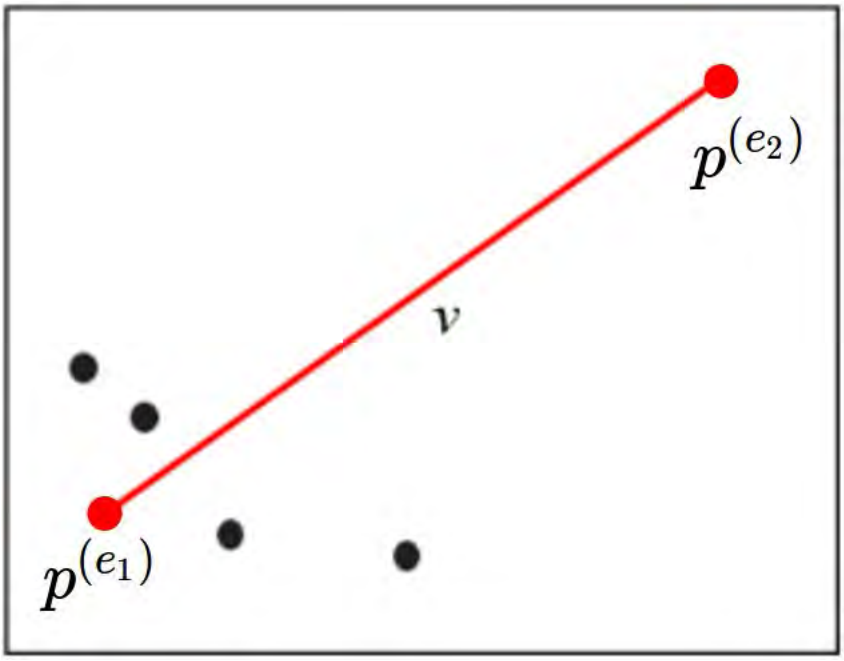


Figure 2.8: $p^{(e_1)}$ and $p^{(e_2)}$ are extreme poles and v is the vector core [10].

Definition 2.10. (The projected order list on the vector core from the extreme pole) Let $e \in \{e_1, e_2\}$ be an index of the extreme pole and $i_1, i_2, ..., i_m \in \{1, 2, ..., m\}$ such that $0 = d(p^{(e)}, p^{(i_1)}) \leq d(p^{(e)}, p^{(i_2)}) \leq ... \leq d(p^{(e)}, p^{(i_m)})$. The projected order list on the vector core from the extreme pole e of a dataset D is defined by

$$\text{OrdList}(D, e) = \{d(p^{(e)}, p^{(i_1)}), d(p^{(e)}, p^{(i_2)}), ..., d(p^{(e)}, p^{(i_m)})\}.$$

Definition 2.11. (The projected order score on the vector core from the extreme pole) The projected order score on the vector core from the extreme pole e is defined by

$$\text{OF}_e(p^{(k)}) = \begin{cases} d(p^{(e)}, p^{(i_2)}) - d(p^{(e)}, p^{(i_1)}) & \text{if } k = i_1 \\ \frac{(d(p^{(e)},p^{(i_k)}) - d(p^{(e)},p^{(i_{k-1})}))(i_k - 1)}{m-1} + \frac{(d(p^{(e)},p^{(i_{k+1})}) - d(p^{(e)},p^{(i_k)}))(m - i_k)}{m-1} & \text{if } k \in \{i_2, ..., i_{m-1}\} \\ d(p^{(e)}, p^{(i_m)}) - d(p^{(e)}, p^{(i_{m-1})}) & \text{if } k = i_m. \end{cases}$$

Definition 2.12**.** (Weighted minimum consecutive pair of the extreme poles outlier factor)

The weighted minimum consecutive pair of the extreme poles outlier factor is defined by

$$\mathrm{WOF}(p^{(k)}) = \frac{\mathrm{OF}_{e_1}(p^{(k)}) + \mathrm{OF}_{e_2}(p^{(k)})}{2}.$$

It is used to represented WOF anomalous score of each instance in a dataset.

Example 2.4**.** Consider a dataset D in Example 2.3. WOF anomalous score of each instance in D based on the Euclidean distance can be computed as follows.

Step 1: Construct the matrix of distance.

$$M = \begin{bmatrix} 0.0 & 5.006 & 5.548 & 5.491 & 6.576 \\ 5.006 & 0.0 & 1.211 & 1.448 & 1.672 \\ 5.548 & 1.211 & 0.0 & 2.553 & 1.21 \\ 5.491 & 1.448 & 2.553 & 0.0 & 2.415 \\ 6.576 & 1.672 & 1.21 & 2.415 & 0.0 \end{bmatrix}.$$

Step 2: Find the extreme poles.

We will get $p^{(1)}$ and $p^{(5)}$ are the extreme poles.

Step 3: Generate the projected order list on the vector core based on the extreme poles.

$$\begin{aligned} \mathrm{OrdList}(D, p^{(1)}) &= \{d_2(p^{(1)}, p^{(1)}), d_2(p^{(1)}, p^{(2)}), d_2(p^{(1)}, p^{(4)}), d_2(p^{(1)}, p^{(3)}), d_2(p^{(1)}, p^{(5)})\} \\ &= \{0, 5.006, 5.491, 5.548, 6.576\} \\ \mathrm{OrdList}(D, p^{(5)}) &= \{d_2(p^{(5)}, p^{(5)}), d_2(p^{(5)}, p^{(3)}), d_2(p^{(5)}, p^{(2)}), d_2(p^{(5)}, p^{(4)}), d_2(p^{(5)}, p^{(1)})\} \\ &= \{0, 1.21, 1.672, 2.415, 6.576\} \end{aligned}$$

Step 4: Compute the projected order score of each instance on each side of the vector core.

$$
\begin{aligned}
\mathrm{OF}_{p^{(1)}}(p^{(1)}) &= \frac{(d_2(p^{(1)}, p^{(2)}) - d_2(p^{(1)}, p^{(1)}))(5-1)}{5-1} \\
&= \frac{(5.006-0)(4)}{4} \\
&= 5.006 \\
\mathrm{OF}_{p^{(1)}}(p^{(2)}) &= \frac{((d_2(p^{(1)}, p^{(2)}) - d_2(p^{(1)}, p^{(1)}))(2-1) + d(p^{(1)}, p^{(4)}) - d(p^{(1)}, p^{(2)}))(5-2)}{5-1} \\
&= \frac{(5.006-0)(1) + (5.491-5.006)(3)}{4} \\
&= 1.615 \\
\mathrm{OF}_{p^{(1)}}(p^{(3)}) &= 0.3 \\
\mathrm{OF}_{p^{(1)}}(p^{(4)}) &= 0.271 \\
\mathrm{OF}_{p^{(1)}}(p^{(5)}) &= 1.028
\end{aligned}
$$

Similarly apply this computation to the rest of instances.

$$
\begin{aligned}
\mathrm{OF}_{p^{(5)}}(p^{(1)}) &= 4.161 \\
\mathrm{OF}_{p^{(5)}}(p^{(2)}) &= 0.602 \\
\mathrm{OF}_{p^{(5)}}(p^{(3)}) &= 0.649 \\
\mathrm{OF}_{p^{(5)}}(p^{(4)}) &= 1.598 \\
\mathrm{OF}_{p^{(5)}}(p^{(5)}) &= 1.21
\end{aligned}
$$

Step 5: Compute the weighted minimum consecutive pair of the extreme poles outlier factor for each instance.

$$
\begin{aligned}
\mathrm{WOF}(p^{(1)}) &= \frac{\mathrm{OF}_{p^{(1)}}(p^{(1)}) + \mathrm{OF}_{p^{(5)}}(p^{(1)})}{2} \\
&= \frac{5.006 + 4.161}{2} \\
&= 4.583
\end{aligned}
$$

Similarly compute for $\text{WOF}(p^{(2)}), \text{WOF}(p^{(3)}), \text{WOF}(p^{(4)})$ and $\text{WOF}(p^{(5)})$ which are $\text{WOF}(p^{(2)}) = 1.109, \text{WOF}(p^{(3)}) = 0.474, \text{WOF}(p^{(4)}) = 0.934$ and $\text{WOF}(p^{(5)}) = 1.119$.

2.6 Detection threshold

2.6.1 Adjusted boxplot

When the criterion based on the boxplot is used to detect outlier on a dataset which has a skew distribution, many values are often incorrectly detected as outliers. The adjustment of the boxplot for a skew distribution is presented by Hubert, et al. [12] in 2008 to generalize the threshold for detecting the outliers. It uses the medcouple (MC) to measure the skewness of univariate data distribution which is defined as follows.

Definition 2.13**.** (Medcouple [17])
Let $F = \{x_1, x_2, ..., x_m\}$ be an ordered of univariate distribution and M_F be the median of F. Define the subsets of F, $X^- = \{x_i \in F | x_i < M_F\}$ and $X^+ = \{x_j \in F | x_j > M_F\}$. The medcouple of F is defined as

$$MC(F) = \text{Median of } H(X^-, X^+)$$

where $H(X^-, X^+)$ is given by

$$\left\{ \frac{(x_j - M_F) - (M_F - x_i)}{x_j - x_i} \mid \forall x_i \in X^-, \forall x_j \in X^+ \right\}.$$

The value of medcouple always lies between -1 and 1. A distribution which is skew to the right has a positive medcouple (positive skew), whereas the medcouple has a negative value for a left skewed distribution (negative skew). Moreover, a symmetric distribution has a zero medcouple. See Figure 2.9.

The threshold for detecting the outliers from the adjusted boxplot based on the medcouple is proposed as

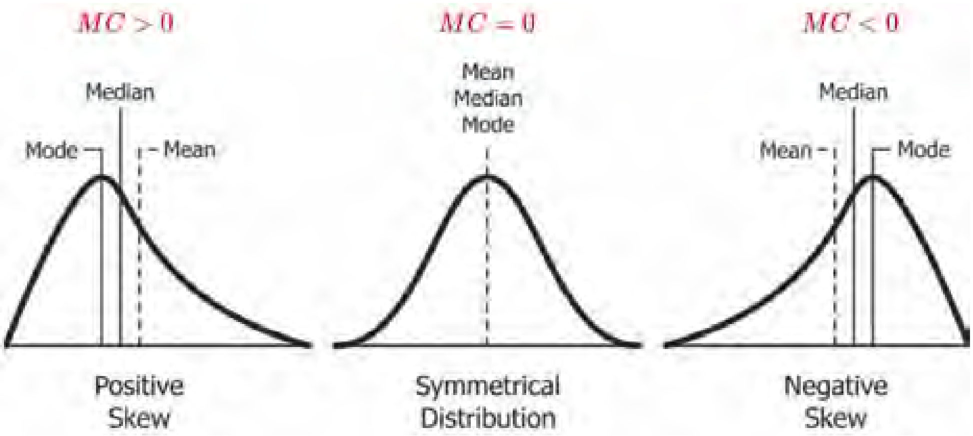


Figure 2.9**:** MC of positive skew, symmetric distribution, and negative skew. source: www.quora.com/What-does-SKEWED-DISTRIBUTION-mean

- for $MC \geq 0$, $[Q_1 - 1.5e^{-4MC}IQR; Q_3 + 1.5e^{3MC}IQR]$
- for $MC < 0$, $[Q_1 - 1.5e^{-3MC}IQR; Q_3 + 1.5e^{4MC}IQR]$.

Example 2.5**.** From Example 2.3 and 2.4, LOF and WOF anomalous score of $p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}$ and $p^{(5)}$ are $2.801, 1.05, 0.971, 1.203, 0.859$ and $4.583, 1.109, 0.474, 0.934, 1.119$, respectively. The threshold from the adjusted boxplot for detecting the outliers of these anomalous scores can be generated as follows.

For $S_1 = \{2.801, 1.05, 0.971, 1.203, 0.859\}$.

Step 1: Sort S_1 as $S_1^{'} = \{0.859, 0.971, 1.05, 1.203, 2.801\}$. Let $M_{S_1^{'}}$ as the median of $S_1^{'}$ which is 1.05, Q_1 as the first-quartile of $S_1^{'}$ which is 0.971, Q_3 as the third-quartile of $S_1^{'}$ which is 1.203, and $IQR = Q_3 - Q_1 = 0.232$.

Step 2: Let $X^- = \{x_i \in S_1^{'} | x_i < M_{S_1^{'}}\}$ and $X^+ = \{x_j \in S_1^{'} | x_j > M_{S_1^{'}}\}$. Then,

$$X^- = \{0.859, 0.971\} \text{ and } X^+ = \{1.203, 2.801\}.$$

Step 3: Construct $H(X^{-},X^{+})=\left\{\dfrac{(x_j-M_F)-(M_F-x_i)}{x_j-x_i}\mid\forall x_i\in X^{-},\forall x_j\in X^{+}\right\}$

$$=\{\frac{(1.203-1.05)-(1.05-0.859)}{1.203-0.859},\frac{(1.203-1.05)-(1.05-0.971)}{1.203-0.971},$$
$$\frac{(2.801-1.05)-(1.05-0.859)}{2.801-0.859},\frac{(2.801-1.05)-(1.05-0.971)}{2.801-0.971}\}$$
$$=\{-0.11,0.319,0.803,0.914.\}$$

Step 4: Compute $MC(S_1^{'})$ is the median of $H(X^{-},X^{+})$.

$$MC(S_1^{'})=Median(\{-0.11,0.319,0.803,0.914\})=0.561$$

Step 5: Since $MC(S_1^{'})\geq 0$, then the threshold is computed by

$$Q_3+1.5e^{3MC}IQR=1.203+(1.5)(e^{30.561})(0.232)=3.077$$

Note. In this thesis, the upper threshold is applied only because any outlier must have a large score.

The threshold for detecting the outliers of S_1 is equal to 3.007 i.e., an element in S_1 is greater than 3.007 will be detected as an outlier while an element is not greater than 3.007 will be detected as a normal. Therefore, $p^{(1)}$, $p^{(2)}$, $p^{(3)}$, $p^{(4)}$, and $p^{(5)}$ are the normal instances with respect to LOF anomalous scores.

Similarly perform for WOF scores as $S_2=\{4.583,1.109,0.474,0.934,1.119\}$. The final result will show that $p^{(1)}$ is detected as an outlier while $p^{(2)},p^{(3)},p^{(4)}$ and $p^{(5)}$ as the normal instances where the value of threshold is equal to 1.324.

2.7 Performance measurements

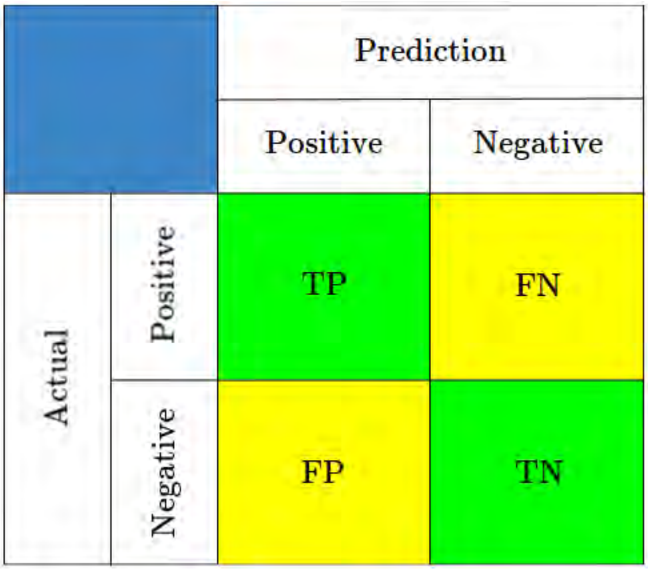
The number of correct and incorrect predictions of the algorithm can be summarized with count values and divide by each class in the confusion matrix ([18], [19], [20]), see Figure 2.10. The entries in the confusion matrix have the following meaning.

- True positive (TP) is the number of correct positive predictions (actual is positive, predicted as positive).

- False positive (FP) is the number of incorrect positive prediction (actual is negative, predicted as positive).

- True negative (TN) is the number of correct negative prediction (actual is negative, predicted as negative).

- False negative (FN) is the number of incorrect negative prediction (actual is positive, predicted as negative).



		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Figure 2.10**:** The confusion matrix.

This thesis focus on the performance of the algorithm for detecting the outliers in a dataset. In a confusion matrix, the positive class represents as the class of outliers and the negative class represents as the class of normal instances where the size of outlier class will be very much smaller than the size of normal class. Since precision is a measure for computing the percentage of the number of detected instances (prediction) which are outliers (actual), recall is a measure for computing the percentage of the number of outliers (actual) which are detected (prediction), and F_1-measure is the harmonic mean of precision and recall. Consequently, these three measurements are used to evaluate the performance of the algorithm for detecting the outliers.

2.7.1 Precision

The precision (also called positive predictive value) is calculated as the number of correct positive predictions (TP) divided by the total number of positive predictions (TP + FP). The largest precision is 1, while the smallest is 0.

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}}.$$

This measure can answer the question that how many predicted positives are actual positives. For example, let $D = \{p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}, p^{(5)}\}$ be a dataset such that $p^{(1)}, p^{(2)}$ are positives and $p^{(3)}, p^{(4)}, p^{(5)}$ are negatives. If the algorithm predicts $p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}, p^{(5)}$ are positives, then Precision $= \frac{2}{2+3} = \frac{2}{5}$ which is 40%.

2.7.2 Recall

The recall (also called sensitivity or true positive rate) is calculated as the number of correct positive predictions (TP) divided by the total number of positives. The best recall is 1, while the worst is 0.

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}.$$

This measure can answer the question that how many actual positives are predicted as positives. From the previous example, Recall $= \frac{2}{2+0} = 1$ which is 100%.

2.7.3 F_1-measure

The F_1-measure is the harmonic mean of precision and recall. So its value lies between 0 and 1. It is calculated as

$$F_1\text{-measure} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Example 2.6**.** Let D be a dataset having 100 instances which are $p^{(1)}, p^{(2)}, ...,$ and $p^{(100)}$ where $p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}$, and $p^{(5)}$ are the outliers and the rest are normal. Assume two anomaly detection algorithms are used which are $Alg1$ and $Alg2$ to detect the outliers in a dataset D. The result from $Alg1$ detects $p^{(1)}, p^{(2)}, ..., p^{(10)}$ as outliers and the result from $Alg2$ detects $p^{(1)}, p^{(3)}, p^{(5)}, p^{(7)}, p^{(9)}$ as outliers. Note that, the positive class instances are labeled as the outliers and the negative class instances are labeled as normals.

Therefore, values of precision, recall, and F_1-measure of $Alg1$ and $Alg2$ are

$$\begin{aligned}
\text{Precision}_{Alg1} &= \frac{5}{5+5} = 0.5 \\
\text{Precision}_{Alg2} &= \frac{3}{3+2} = 0.6 \\
\text{Recall}_{Alg1} &= \frac{5}{5+0} = 1.0 \\
\text{Recall}_{Alg2} &= \frac{3}{3+2} = 0.6 \\
F_1\text{-measure}_{Alg1} &= 2 \cdot \frac{(0.5 \times 1.0)}{(0.5 + 1.0)} = 0.667 \\
F_1\text{-measure}_{Alg2} &= 2 \cdot \frac{(0.6 \times 0.6)}{(0.6 + 0.6)} = 0.600
\end{aligned}$$

From these computations, $Alg2$ shows better precision than $Alg1$ while $Alg1$ shows better recall and F_1-measure than $Alg2$. So $Alg1$ shows better overall performances with respect to $Alg2$.

CHAPTER III

C-ANOMALOUS ASSEMBLAGE DETECTION USING NEAREST NEIGHBOR DISTANCE

This chapter covers the definition of distance-based outlier, C-anomalous assemblage, k^{th}-nearest neighbor index and k^{th}-nearest neighbor distance. In addition, a new anomaly detection algorithm for detecting C-anomalous assemblages called CND is presented.

3.1 Preliminaries

This thesis focuses on a dataset with continuous-valued attributes where the outlierness of an instance is considered based on the distance between two instances. The definition of outlier corresponding with the distance-based approach are defined as follows.

Definition 3.1. (Distance-based outlier)
An outlier is an instance that lies farthest away from majority instances of a dataset.

To complete this definition, a majority must be defined. However, this thesis defines the negate of majority instead. It uses a user's defined parameter C which represents the percentage of non-majority clusters. A group of outliers having the number of neighbour instances less than or equal to C percent of the total number of instances in a dataset is defined as the anomalous assemblage. An anomalous assemblage should be small and lie far away from other clusters in a dataset.

Definition 3.2. (C-anomalous assemblage)
The C-anomalous assemblage is an anomalous assemblage having the number of instances less than or equal to C percent of the total number of instances in a dataset.

Example 3.1. Let D be a dataset in $\mathbb{R}^2$ having 100 instances. See Figure 3.1.

The 5-anomalous assemblage is an anomalous assemblages having the number of instances less than or equal to 5 percent of the total instances ($\leq \frac{5}{100} \times 100 = 5$). Then, O_1, O_2, O_3, O_4 and O_5 are the 5-anomalous assemblages.

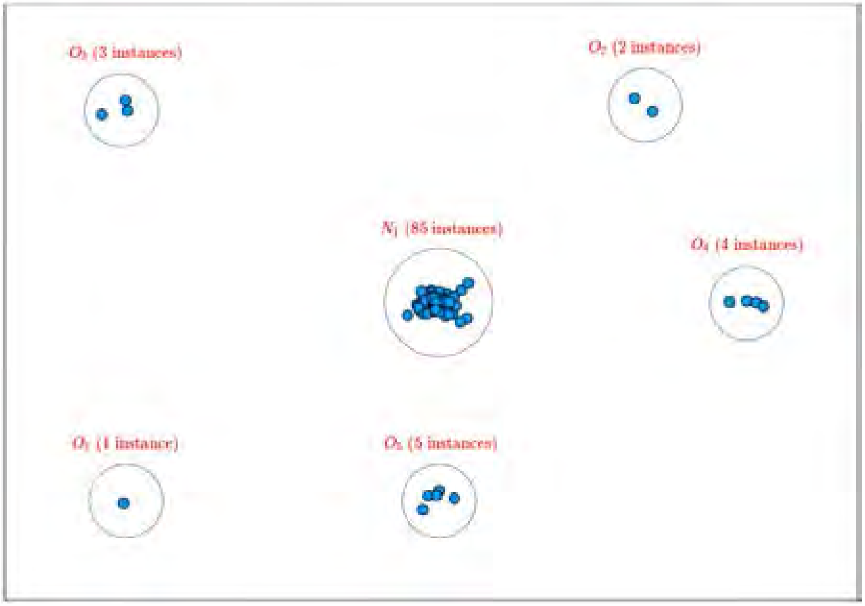


Figure 3.1**:** The 5-anomalous assemblages on a dataset $D \subseteq \mathbb{R}^2$ having 100 instances.

Next, the concept of k-nearest neighbors is used to represent the proximity between an instance and its neighbors. The distance of each instance into its neighbors provides the important information to decide whether an instance should be an outlier or a normal instance. See the following definition.

Definition 3.3. (k^{th}-nearest neighbor index)
For $i \in \{1, 2, ..., m\}$, $N_i(k)$ is defined as the index of the k^{th}-nearest neighbor of $p^{(i)}$ which is recursively defined for each $k \in \{0, 1, 2, ..., m-1\}$ as

$$N_i(k) = \underset{j \in \{1,...,m\} \setminus \{N_i(0), N_i(1), ..., N_i(k-1)\}}{\operatorname{argmin}} \{d(p^{(i)}, p^{(j)})\}$$

where $N_i(0) = i$.

Definition 3.4. (k^{th}-nearest neighbor distance)

For $i \in \{1, 2, ..., m\}$ and $k \in \{0, 1, .., m-1\}$, the k^{th}-nearest neighbor distance of $p^{(i)}$ is defined as $ND(i, k)$ which is the distance between $p^{(i)}$ and $p^{(N_i(k))}$.

Example 3.2. Let $p^{(1)}, p^{(2)}$ and $p^{(3)}$ are instances in $\mathbb{R}^2$. See Figure 3.2.

- The 1^{st}-nearest neighbor index of $p^{(1)}$ is represented by $N_1(1) = 2$ and the 1^{th}-nearest neighbor distance of $p^{(1)}$ is represented $ND(1, 1) = d_2(p^{(1)}, p^{(2)}) = 1$.
- The 2^{nd}-nearest neighbor index of $p^{(1)}$ is represented by $N_1(2) = 3$ and the 2^{nd}-nearest neighbor distance of $p^{(1)}$ is represented $ND(1, 2) = d_2(p^{(1)}, p^{(3)}) = 2$.

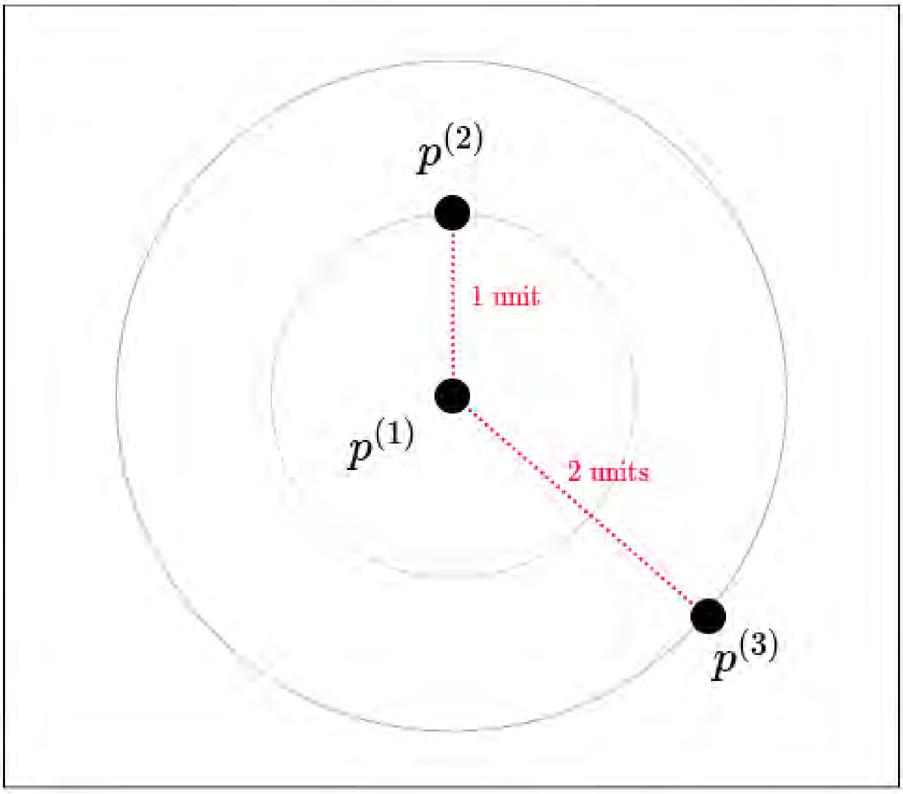


Figure 3.2**:** The k^{th}-nearest neighbor index and distance of $p^{(1)}$ for $k = 1, 2$ based on the Euclidean distance in $\mathbb{R}^2$.

Basic idea of CND

Let $D \subseteq \mathbb{R}^n$ be a dataset having m instances with O as an anomalous assemblage having t instances and N is a normal cluster having $m - t$ instances. Observe that

• For any $k < t$, the k^{th}-nearest neighbor distance of the instances in O and N are not significantly difference.

• For any $k \in \{t, t+1, ..., (m-t-1)\}$, the k^{th}-nearest neighbor distance of the instances in O and N are significantly difference where the distances for instances in O are high and the distances for instances in N are small.

From Figure 3.3, if the value of index k covers the size of an anomalous assemblage, the k^{th}-nearest neighbor distance of these outliers are large which are significantly larger than the k^{th}-nearest neighbor distance of any normal instance. Consequently, the k^{th}-nearest neighbor distance of each instance can be used to represent an anomalous score where the index k should be selected to cover the size of any C-anomalous assemblage in a dataset.

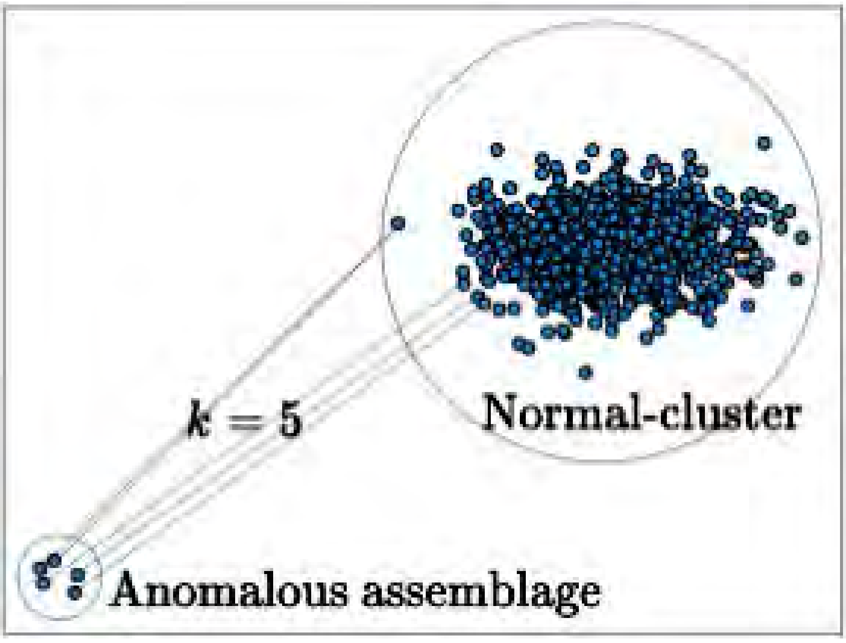


Figure 3.3: Basic idea of CND

.

3.2 CND algorithm

A new anomaly detection algorithm is called CND which is presented for effectively detecting the C-anomalous assemblages. The input of the algorithm composes of a finite dataset $D \subseteq \mathbb{R}^n$ and an associated parameter C. Then, the k^{th} index is computed from the floor function of C percent of the total number of instances and the k^{th}-nearest neighbor distance of each instance is computed as an anomalous score. Next, the scores are split by the threshold based on the adjusted boxplot using only the upper threshold because any outlier will only have a large score. Finally, the output reports the set of outliers. The algorithm is shown next.

```
INPUT: A dataset $D \subseteq \mathbb{R}^n$ with $m$ instances and a parameter C.
Step 1. Compute $k = \left\lfloor \frac{C}{100} \times m \right\rfloor$.
Step 2. Construct $S = \{ND(i,k) \mid i = 1,2,...,m\}$
Step 3. Compute $Q_1$ = first-quartile($S$), $Q_3$ = third-quartile($S$), $MC(S)$,
        and $IQR = Q_3 - Q_1$.
Step 4. If $MC(S) \geq 0$
          Let  threshold $= Q_3 + 1.5e^{3MC(S)}IQR$
        Else
          Let  threshold $= Q_3 + 1.5e^{4MC(S)}IQR$
        End
Step 5. $O = [\,]$
        For $i = 1,2,...,m$
          If $ND(i,k) >$ threshold
            $O \leftarrow O \cup \{i\}$
          End
        End
        Return $O$
OUTPUT: $O$ is the set of detected outliers.
```

Example 3.3**.** Let $D \subseteq \mathbb{R}^2$ be a dataset having 100 instances. In Figure 3.4, let parameter $C = 5$ for CND (based on the Euclidean distance), then $k = \left\lfloor \frac{5}{100} \times 100 \right\rfloor = 5$ is used to compute a CND anomalous score of each instance.

For $p^{(1)} = (-2.025, -2.0219)$, the 5^{th}-nearest neighbor index of $p^{(1)}$ is $p^{(6)}$ i.e., $N_1(5) = 6$. Then the 5^{th}-nearest neighbor distance of $p^{(1)}$ is the distance between $p^{(1)}$ and $p^{(6)}$ which is equal to 12.798 i.e., $ND(1, 5) = 12.798$. So, CND anomalous score of $p^{(1)}$ is 12.798. Similarly perform for $p^{(2)}, p^{(3)}, p^{(4)}, p^{(5)}, p^{(6)}$, the CND anomalous scores are 11.728, 11.79, 11.659, 10.097, 1.203, respectively.

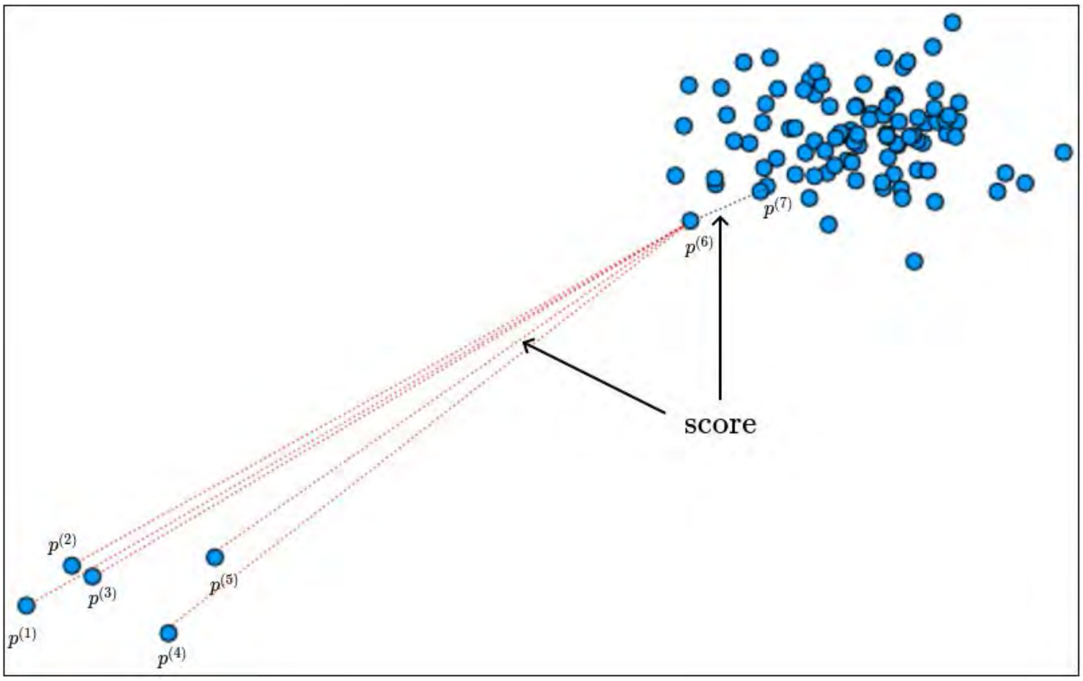


Figure 3.4**:** CND anomalous scores of $p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}, p^{(5)}$, and $p^{(6)}$.

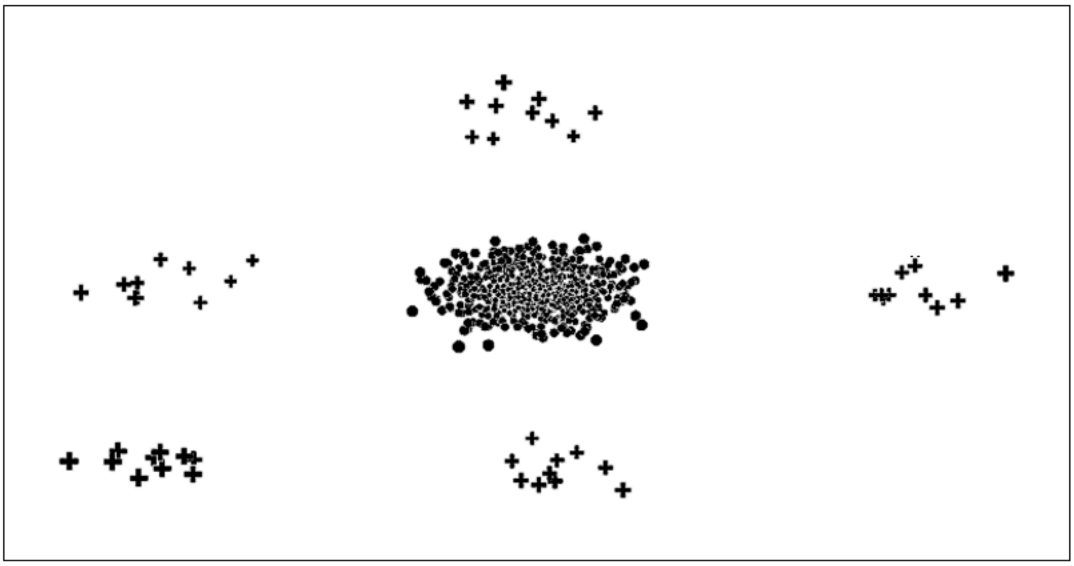


Figure 3.5: A dataset on two dimensional space with one cluster having 1000 instances is performed by CND using $C = 5$. An outlier is labeled by a plus symbol, while a normal instance is labeled by a circle symbol.

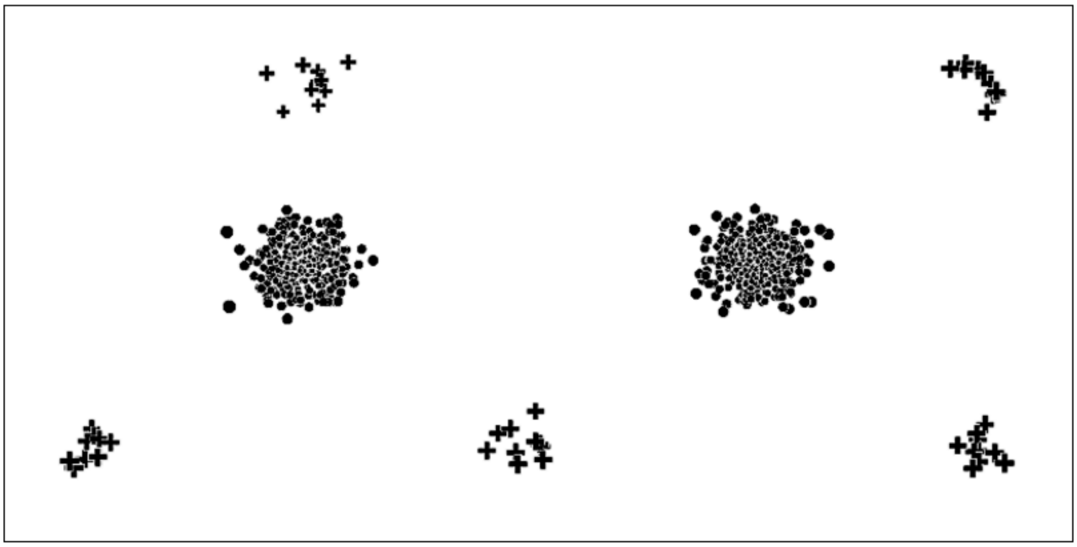


Figure 3.6: A dataset on two dimensional space with two clusters having 1000 instances is performed by CND using $C = 5$. An outlier is labeled by a plus symbol, while a normal instance is labeled by a circle symbol.

3.3 Time complexity

Time complexity of CND is analyzed and split into two parts. The first part is the computation of the anomalous scores using the k^{th}-nearest neighbor distance and the second part is the computation of the medcouple for generating the threshold.

Given a dataset D having n instances.

- The first part: the k^{th}-nearest neighbor distance of each instance is computed where it has the time complexity $O(n^2)$.

- The second part: the n anomalous scores are used to compute the medcouple where it has the time complexity $O(n^2)$.

Therefore, the overall time complexity of CND is $O(n^2) + O(n^2) = O(n^2)$.

CHAPTER IV

EXPERIMENTAL RESULTS

This chapter covers the experimental result for the performance of CND detecting the outliers based on precision, recall, and F_1-measure on two types of datasets which are synthetic and real-world datasets comparing with WOF and LOF. Moreover, the suitable parameter C of CND algorithm will be investigated. Note that

- All experiments are implemented via the Julia programming language version 0.5.
- The distance between any two instances is represented by the Euclidean distance.
- WOF and LOF are scoring algorithm and their papers did not suggest any threshold for detecting the outliers. It is left to the reader to decide the threshold themselves. All experiments in this thesis use the adjusted boxplot to generate the threshold for detecting the outliers so that they could be compared using precision, recall and F_1-measure.
- The positive class instances are labeled as the outliers and the negative class instances are labeled as the normal instances.
- Each dataset is deliberately designed to have the anomalous assemblages where the number of outliers is set to 5 % of total number of instances in a dataset.

4.1 Synthetic dataset

Five collections of synthetic datasets which are the collection having one normal cluster, two normal clusters, three normal clusters, four normal clusters, and five normal clusters on the two-dimensional space are randomly generated. Each dataset is constructed to contain 1000 instances with 950 normals and 50 outliers (5 % outliers). The detail of each collection is shown as follows.

Collection 1: One cluster

The first collection contains 5 subcollections which are the subcollection of one normal cluster with 1-5 anomalous assemblages generated by the following descriptions.

- Collection 1.1 contains 10 datasets where each dataset is randomly generated a normal cluster (N_1) having 950 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, an anomalous assemblage having 50 instances are randomly generated with the same covariance matrix along 8 different centroid locations, see in Figure 4.1, around a normal cluster about 100 units apart.

- Collection 1.2 contains 10 datasets where each dataset is randomly generated a normal cluster (N_1) having 950 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, two anomalous assemblages having 25 instances for each assemblage are randomly generated with the same covariance matrix along 8 different centroid locations, see in Figure 4.1, around a normal cluster about 100 units apart.

- Collection 1.3 contains 10 datasets where each dataset is randomly generated a normal cluster (N_1) having 950 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, three anomalous assemblages having 16, 17, 17 instances are randomly generated with the same covariance matrix along 8 different centroid locations, see in Figure 4.1, around a normal cluster about 100 units apart.

- Collection 1.4 contains 10 datasets where each dataset is randomly generated a normal cluster (N_1) having 950 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, four anomalous assemblages having 12, 12, 13, 13 instances are randomly generated with the same covariance matrix along 8 different centroid locations, see in Figure 4.1, around a normal cluster about 100 units apart.

- Collection 1.5 contains 10 datasets where each dataset is randomly generated a normal cluster (N_1) having 950 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, five anomalous assemblages having 10 instances for each assemblage are randomly generated with the same covariance matrix along 8 different centroid locations, see in Figure 4.1, around a normal cluster about 100 units apart.

The conceptual diagram to generate the collection 1 of synthetic datasets is displayed in Figure 4.1.

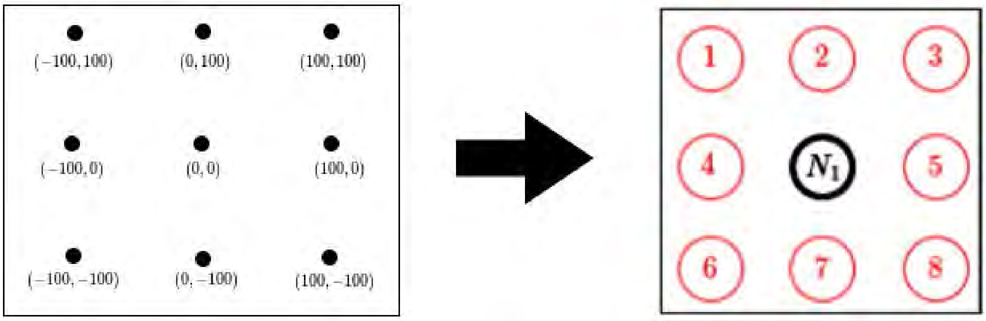


Figure 4.1: The model to generate the collection 1 of synthetic datasets. The normal cluster is labeled by the circle N_1, while the anomalous assemblages are labeled by the circle number 1-8.

Collection 2: Two clusters

The second collection contains 5 subcollections which are the subcollection of two normal clusters with 1-5 anomalous assemblages generated by the following descriptions.

- Collection 2.1 contains 10 datasets where each dataset is randomly generated two normal clusters (N_1 and N_2) having 475 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, an anomalous assemblage having 50 instances are randomly generated with the same covariance matrix along 13 different centroid locations, see in Figure 4.2, around the normal clusters about 100 units apart.

- Collection 2.2 contains 10 datasets where each dataset is randomly generated two normal clusters (N_1 and N_2) having 475 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, two anomalous assemblages having 25 instances for each assemblage are randomly generated with the same covariance matrix along 13 different centroid locations, see in Figure 4.2, around the normal clusters abouts 100 units apart.

- Collection 2.3 contains 10 datasets where each dataset is randomly generated two normal clusters (N_1 and N_2) having 475 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, three anomalous assemblages having 16, 17, 17 instances are randomly generated with the same covariance matrix along 13 different centroid locations, see in Figure 4.2, around the normal clusters about 100 units apart.

- Collection 2.4 contains 10 datasets where each dataset is randomly generated two normal clusters (N_1 and N_2) having 475 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, four anomalous assemblages having 12, 12, 13, 13 instances are randomly generated with the same covariance matrix along 13 different centroid locations, see in Figure 4.2, around the normal clusters about 100 units apart.

- Collection 2.5 contains 10 datasets where each dataset is randomly generated two normal clusters (N_1 and N_2) having 475 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, five anomalous assemblages having 10 instances for each assemblage are randomly generated with the same covariance matrix along 13 different centroid locations, see in Figure 4.2, around the normal clusters about 100 units apart.

The conceptual diagram to generate the collection 2 of synthetic datasets is displayed in Figure 4.2.

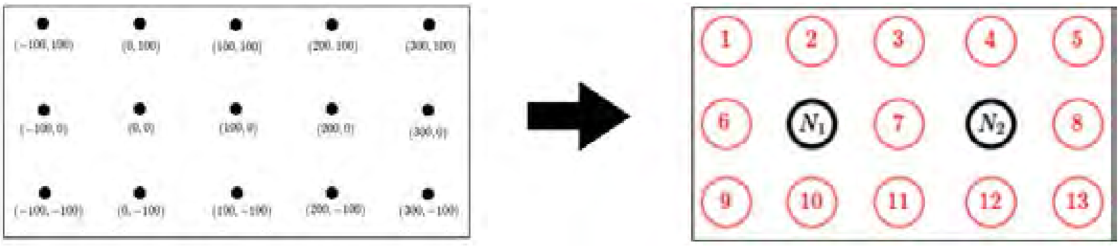


Figure 4.2**:** The model to generate the collection 2 of synthetic datasets. The normal clusters are labeled by the circle N_1 and N_2, while the anomalous assemblages are labeled by the circle number 1-13.

Collection 3: Three clusters

The third collection contains 5 subcollections which are the subcollection of three normal clusters with 1-5 anomalous assemblages generated by the following descriptions.

- Collection 3.1 contains 10 datasets where each dataset is randomly generated three normal clusters (N_1, N_2, and N_3) having 316, 317, and 317 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, an anomalous assemblage having 50 instances are randomly generated with the same covariance matrix along 18 different centroid locations, see in Figure 4.3, around the normal clusters about 100 units apart.

- Collection 3.2 contains 10 datasets where each dataset is randomly generated three normal clusters (N_1, N_2, and N_3) having 316, 317, and 317 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, two anomalous assemblages having 25 instances for each assemblage are randomly generated with the same covariance matrix along 18 different centroid locations, see in Figure 4.3, around the normal clusters abouts 100 units apart.

- Collection 3.3 contains 10 datasets where each dataset is randomly generated three normal clusters (N_1, N_2, and N_3) having 316, 317, and 317 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, three anomalous assemblages having 16, 17, 17 instances are randomly generated with the same covariance matrix along 18 different centroid locations, see in Figure 4.3, around the normal clusters about 100 units apart.

- Collection 3.4 contains 10 datasets where each dataset is randomly generated three normal clusters (N_1, N_2, and N_3) having 316, 317, and 317 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, four anomalous assemblages having 12, 12, 13, 13 instances are randomly generated with the same covariance matrix along 18 different centroid locations, see in Figure 4.3, around the normal clusters about 100 units apart.

- Collection 3.5 contains 10 datasets where each dataset is randomly generated three normal clusters (N_1, N_2, and N_3) having 316, 317, and 317 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, five anomalous assemblages having 10 instances for each assemblage are randomly generated with the same covariance matrix along 18 different centroid locations, see in Figure 4.3, around the normal clusters about 100 units apart.

The conceptual diagram to generate the collection 3 of synthetic datasets is displayed in Figure 4.3.

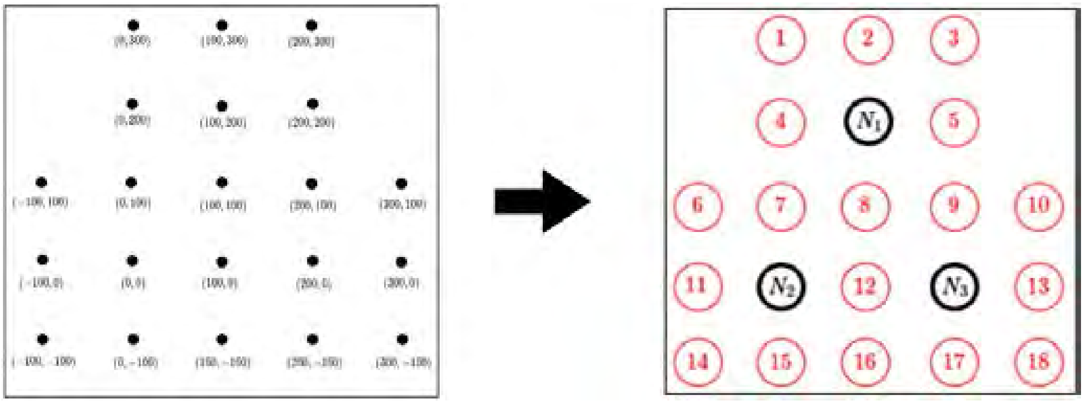


Figure 4.3: The model to generate the collection 3 of synthetic datasets. The normal clusters are labeled by the circle N_1, N_2, and N_3, while the anomalous assemblages are labeled by the circle number 1-18.

Collection 4: Four clusters

The fourth collection contains 5 subcollections which are the subcollection of four normal clusters with 1-5 anomalous assemblages generated by the following descriptions.

- Collection 4.1 contains 10 datasets where each dataset is randomly generated four normal clusters (N_1, N_2, N_3 and N_4) having 237, 237, 238, and 238 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ -200 \end{bmatrix},$ respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, an anomalous assemblage having 50 instances are randomly generated with the same covariance matrix along 23 different centroid locations, see in Figure 4.4, around the normal clusters about 100 units apart.

- Collection 4.2 contains 10 datasets where each dataset is randomly generated four normal clusters (N_1, N_2, N_3 and N_4) having 237, 237, 238, and 238 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ -200 \end{bmatrix},$ respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, two anomalous assemblages having 25 instances for each assemblage are randomly generated with the same covariance matrix along 23 different centroid locations, see in Figure 4.4, around the normal clusters abouts 100 units apart.

- Collection 4.3 contains 10 datasets where each dataset is randomly generated four normal clusters (N_1, N_2, N_3 and N_4) having 237, 237, 238, and 238 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, three anomalous assemblages having 16, 17, 17 instances are randomly generated with the same covariance matrix along 23 different centroid locations, see in Figure 4.4, around the normal clusters about 100 units apart.

- Collection 4.4 contains 10 datasets where each dataset is randomly generated four normal clusters (N_1, N_2, N_3 and N_4) having 237, 237, 238, and 238 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, four anomalous assemblages having 12, 12, 13, 13 instances are randomly generated with the same covariance matrix along 23 different centroid locations, see in Figure 4.4, around the normal clusters about 100 units apart.

- Collection 4.5 contains 10 datasets where each dataset is randomly generated four normal clusters (N_1, N_2, N_3 and N_4) having 237, 237, 238, and 238 instances by the multivariate normal distribution with $\mu = \begin{bmatrix} 100 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, five anomalous assemblages having 10 instances for each assemblage are randomly generated with the same covariance matrix along 23 different centroid locations, see in Figure 4.4, around the normal clusters about 100 units apart.

The conceptual diagram to generate the collection 4 of synthetic datasets is displayed in Figure 4.4.

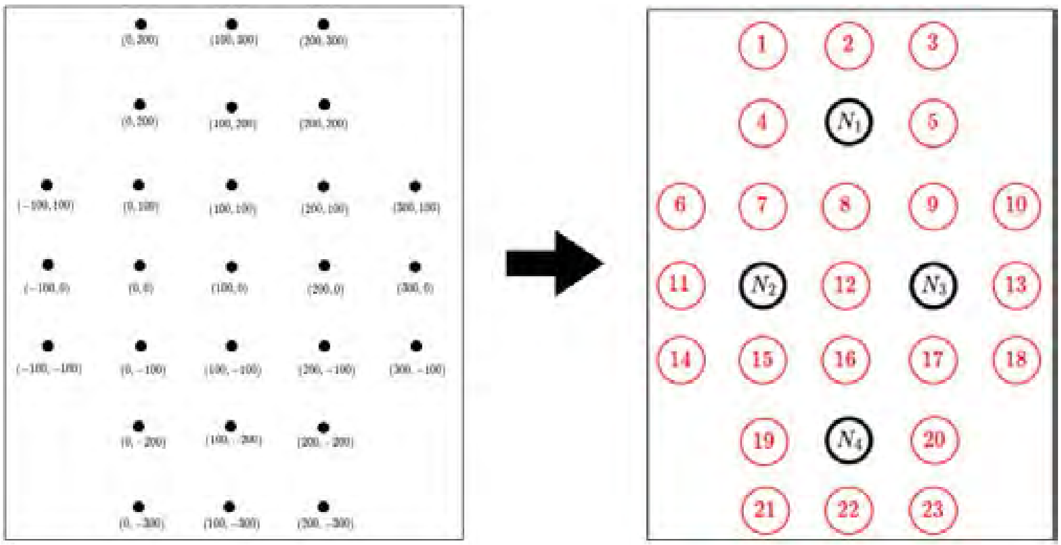


Figure 4.4**:** The model to generate the collection 4 of synthetic datasets. The normal clusters are labeled by the circle N_1, N_2, N_3, and N_4, while the anomalous assemblages are labeled by the circle number 1-23.

Collection 5: Five clusters

The fifth collection contains 5 subcollections which are the subcollection of five normal clusters with 1-5 anomalous assemblages generated by the following descriptions.

- Collection 5.1 contains 10 datasets where each dataset is randomly generated five normal clusters (N_1, N_2, N_3, N_4 and N_5) having 190 instances for each cluster by the multivariate normal distribution with

$$\mu = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 400 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ -200 \end{bmatrix}, \text{ respectively and } \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

for each cluster. Then, an anomalous assemblage having 50 instances are randomly generated with the same covariance matrix along 28 different centroid locations, see in Figure 4.5, around the normal clusters about 100 units apart.

- Collection 5.2 contains 10 datasets where each dataset is randomly generated five normal clusters (N_1, N_2, N_3, N_4 and N_5) having 190 instances for each cluster by the multivariate normal distribution with

$\mu = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 400 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, two anomalous assemblages having 25 instances for each assemblage are randomly generated with the same covariance matrix along 28 different centroid locations, see in Figure 4.5, around the normal clusters abouts 100 units apart.

- Collection 5.3 contains 10 datasets where each dataset is randomly generated five normal clusters (N_1, N_2, N_3, N_4 and N_5) having 190 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 400 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, three anomalous assemblages having 16, 17, 17 instances are randomly generated with the same covariance matrix along 28 different centroid locations, see in Figure 4.5, around the normal clusters about 100 units apart.

- Collection 5.4 contains 10 datasets where each dataset is randomly generated five normal clusters (N_1, N_2, N_3, N_4 and N_5) having 190 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 400 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, four anomalous assemblages having 12, 12, 13, 13 instances are randomly generated with the same covariance matrix along 28 different centroid locations, see in Figure 4.5, around the normal clusters about 100 units apart.

- Collection 5.5 contains 10 datasets where each dataset is randomly generated five normal clusters (N_1, N_2, N_3, N_4 and N_5) having 190 instances for each cluster by the multivariate normal distribution with $\mu = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ 0 \end{bmatrix}, \begin{bmatrix} 400 \\ 0 \end{bmatrix}, \begin{bmatrix} 200 \\ -200 \end{bmatrix}$, respectively and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for each cluster. Then, five anomalous assemblages having 10 instances for each assemblage are randomly generated with the same covariance matrix along 28 different centroid locations, see in Figure 4.5, around the normal clusters about 100 units apart.

The conceptual diagram to generate the collection 5 of synthetic datasets is displayed in Figure 4.5.

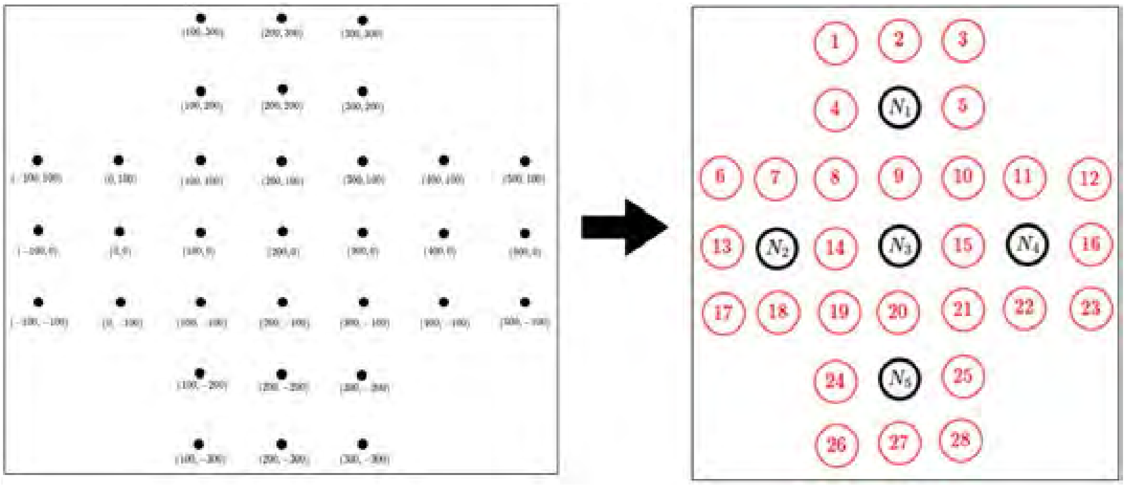


Figure 4.5: The model to generate the collection 5 of synthetic datasets. The normal clusters are labeled by the circle N_1, N_2, N_3, N_4, and N_5, while the anomalous assemblages are labeled by the circle number 1-28.

Synthetic **datasets**	**Normal clusters** **(950 instances)**	**Anomalous assemblages** **(50 instances)**
Collection 1	One cluster	*** One through five assemblages where • 50 for one assemblage. • 25, 25 for two assemblages. • 16, 17, 17 for three assemblages. • 12, 12, 13, 13 for four assemblages. • 10, 10, 10, 10, 10 for five assemblages.
Collection 2	Two clusters (475, 475)	
Collection 3	Three clusters (316, 317, 317)	
Collection 4	Four clusters (237, 237, 238, 238)	
Collection 5	Five clusters (190,190,190,190,190)	

Table 4.1: The summary of the randomly generated synthetic datasets.

To determine the suitable value of the parameter C for synthetic datasets. CND is performed by vary the parameter $C = 1$ to 10 on each dataset in each collection where F_1-measure on each dataset is computed. Next, the average $F1$-measure (AF) of ten generated datasets in each subcollection is computed and the standard deviation of this average $F1$-measure (SDAF) is also computed. Finally, the error bar is plotted by AF ± SDAF of ten generated datasets in each subcollection along $C = 1$ to 10.

Error bars of collection 1

Figure 4.6 displays the error bars of the average F_1-measure along $C = 1$ to 10 on ten generated datasets of each subcollection in collection 1. The important behavior of error bars in each subcollection is shown as follows.

- For collection 1.1: $C = 5$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 6$ to 10.

- For collection 1.2: $C = 3$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 4$ to 10.

- For collection 1.3: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

- For collection 1.4: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

- For collection 1.5: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

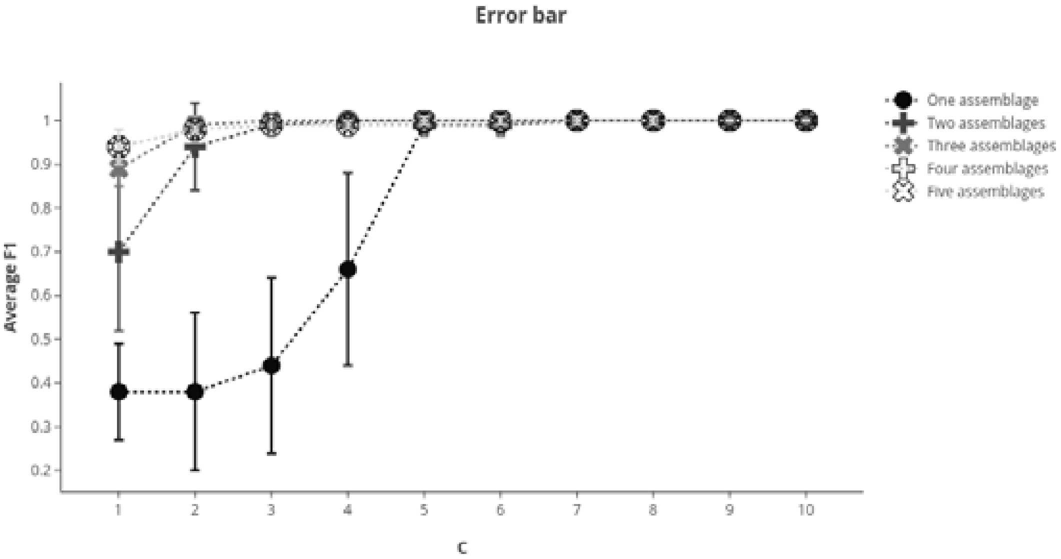


Figure 4.6: The error bars of collection 1 where C along x-axis and the average F_1-measure along y-axis.

Error bars of collection 2

Figure 4.7 displays the error bars of the average F_1-measure along $C = 1$ to 10 on ten generated datasets of each subcollection in collection 2. The important behavior of error bars in each subcollection is shown as follows.

- For collection 2.1: $C = 5$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 6$ to 10.

- For collection 2.2: $C = 3$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 4$ to 10.

- For collection 2.3: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

- For collection 2.4: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

- For collection 2.5: $C = 2$ shows the best average $F1$-measure is about 1.0 and a

small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

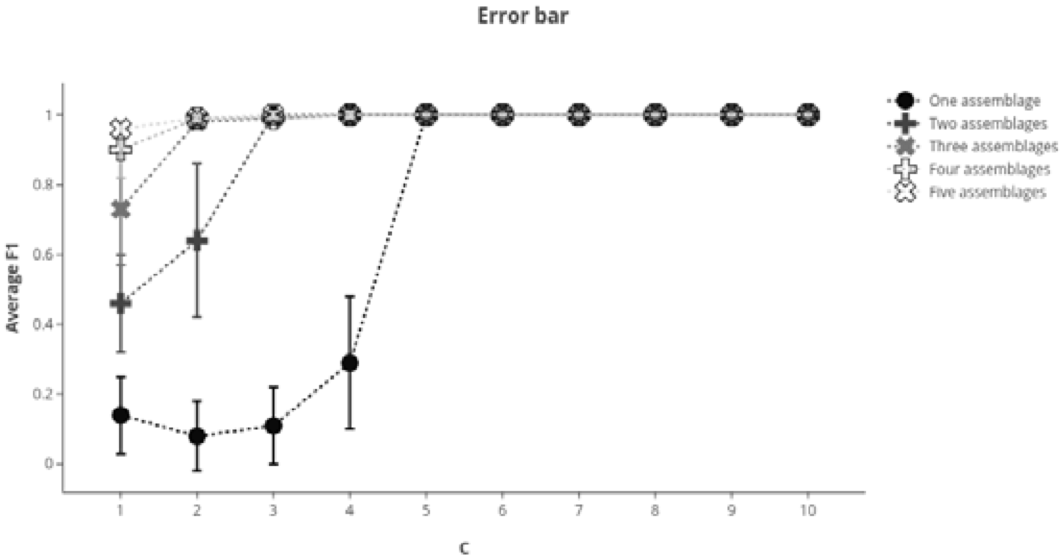


Figure 4.7**:** The error bars of collection 2 where C along x-axis and the average F_1-measure along y-axis.

Error bars of collection 3

Figure 4.8 displays the error bars of the average F_1-measure along $C = 1$ to 10 on ten generated datasets of each subcollection in collection 3. The important behavior of error bars in each subcollection is shown as follows.

• For collection 3.1: $C = 5$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that the average $F1$-measure is stable about 1.0 from $C = 6$ to 10.

• For collection 3.2: $C = 3$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measures is stable about 1.0 from $C = 4$ to 10.

• For collection 3.3: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measures is stable about 1.0 from $C = 3$ to 10.

• For collection 3.4: $C = 2$ shows the best average $F1$-measure is about 1.0 and a

small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

• For collection 3.5: $C = 1$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 2$ to 10.

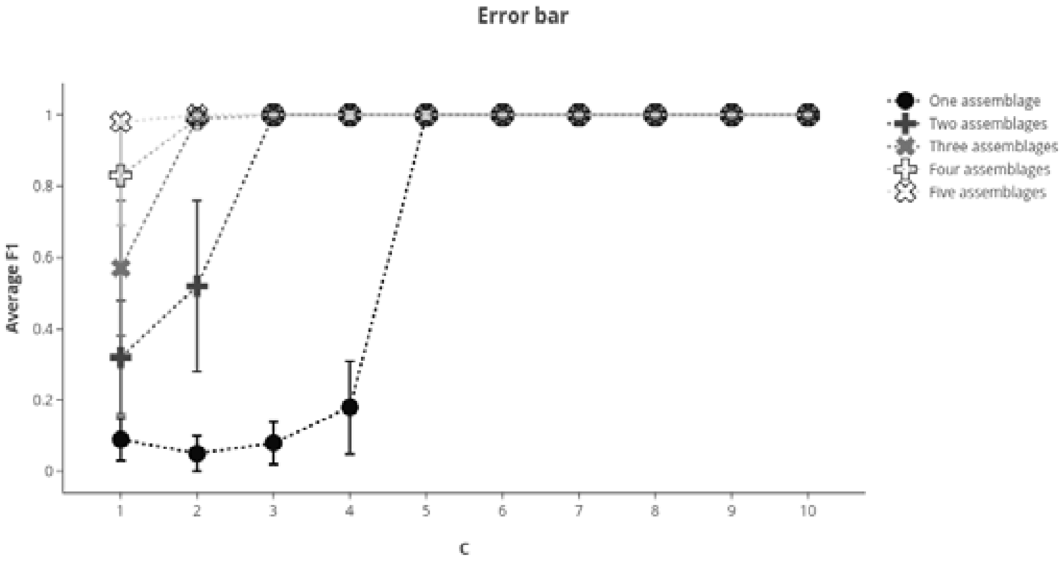


Figure 4.8**:** The error bars of collection 3 where C along x-axis and the average F_1-measure along y-axis.

Error bars of collection 4

Figure 4.9 displays the error bars of the average F_1-measure along $C = 1$ to 10 on ten generated datasets of each subcollection in collection 4. The important behavior of error bars in each subcollection is shown as follows.

• For collection 4.1: $C = 5$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 6$ to 10.

• For collection 4.2: $C = 3$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 4$ to 10.

• For collection 4.3: $C = 2$ shows the best average $F1$-measure is about 1.0 and a

small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

• For collection 4.4: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.

• For collection 4.5: $C = 1$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 2$ to 10.

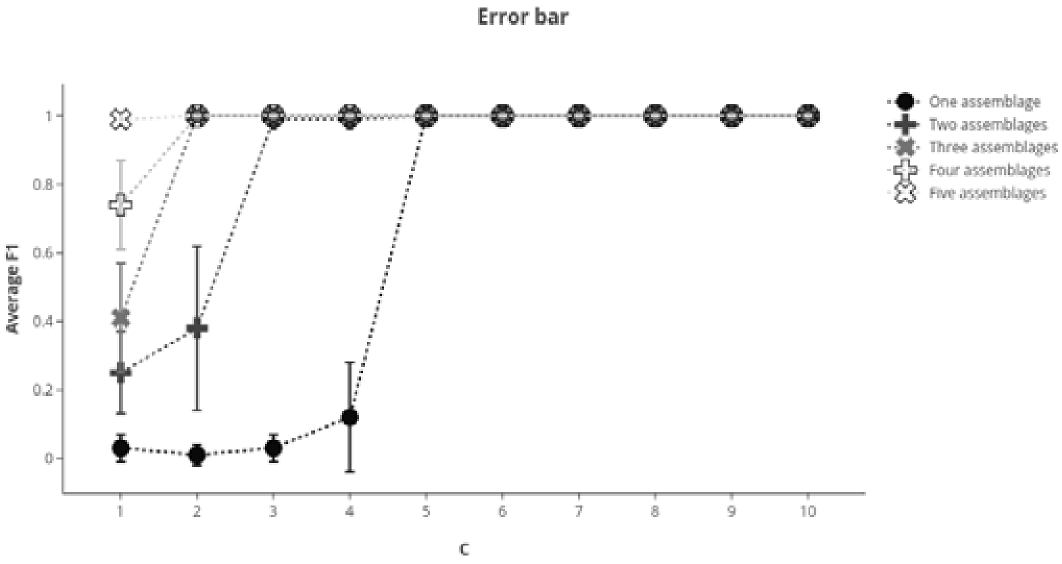


Figure 4.9**:** The error bars of collection 4 where C along x-axis and the average F_1-measure along y-axis.

Error bars of collection 5

Figure 4.10 displays the error bars of the average F_1-measure along $C = 1$ to 10 on ten generated datasets of each subcollection in collection 5. The important behavior of error bars in each subcollection is shown as follows.

• For collection 5.1: $C = 5$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 6$ to 10.

• For collection 5.2: $C = 3$ shows the best average $F1$-measure is about 1.0 and a

small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 4$ to 10.

- For collection 5.3: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.
- For collection 5.4: $C = 2$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 3$ to 10.
- For collection 5.5: $C = 1$ shows the best average $F1$-measure is about 1.0 and a small standard deviation. After that, the average $F1$-measure is stable about 1.0 from $C = 2$ to 10.

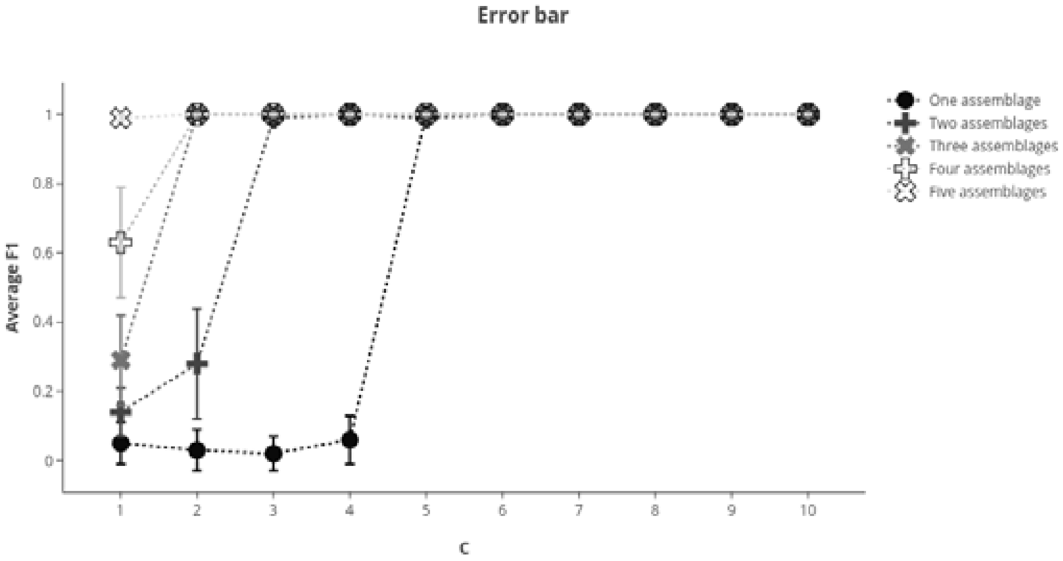


Figure 4.10**:** The error bars of collection 5 where C along x-axis and the average F_1-measure along y-axis.

Discussion of error bars on synthetic datasets

The results from Figure 4.6 - 4.10 show the suitable value of parameter C for each size of the anomalous assemblage where if the value of $C\%$ covers the size of each anoamlous assemblage then CND will have the good performance for detecting these outliers as follows.

$C = 5$ is the suitable value of the parameter for a dataset having one anomalous assemblage (50 instances), $C = 3$ is the suitable value of the parameter for a dataset having two anomalous assemblages (25 instances for each assemblage), $C = 2$ is the suitable value of the parameter for a dataset having three anomalous assemblages (16,17, and 17 instances) and four anomalous assemblages (12,12,13, and 13 instances), $C = 1$ is the suitable value of the parameter for a dataset having five anomalous assemblages (10 instances for each assemblage).

Remark. Since each dataset is constructed having the number of outliers equal to 5%, CND with $C = 5$ will guarantee the best performance for these datasets.

To compare the performance of CND with WOF and LOF on synthetic datasets, the parameter C is set to 5 for CND and $k = \left\lfloor \frac{5}{100} \times m \right\rfloor$ for LOF. The performance results are shown in Figure 4.11 - 4.25.

Discussion of the performance on synthetic datasets

For overall performance on synthetic datasets, the results show that CND is better than WOF and LOF based on precision, recall, and F_1-measure for all collections of synthetic datasets (WOF is worst). Moreover, there are important observations that can conclude from the performance graphs of synthetic datasets as follows.

1.) CND is better than WOF and never worse than LOF based on precision and F_1-measure for all datasets (CND and LOF have the same precision and F_1-measure for some datasets).

2.) CND is similar to LOF which never worse than WOF based on recall (CND, WOF, and LOF have the same recall for some datasets).

3.) LOF is better than WOF based on precision and F_1-measure for all datasets.

4.) CND and LOF can correctly identify actual outliers for all datasets since recall is 100%. However, there are some normal instances are detected as the outliers for some datasets since precision is less than 100%.

5.) If the value of threshold is raised, then LOF may have better performance for precision and F_1-measure.

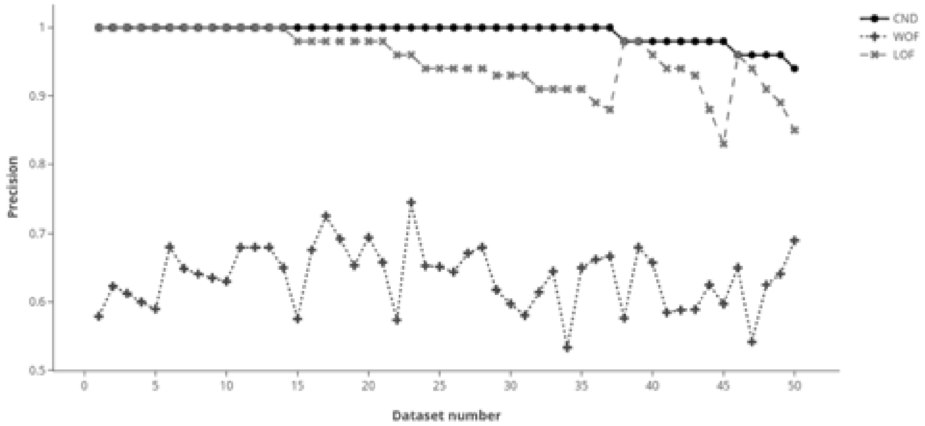


Figure 4.11**:** The precision on collection 1 of synthetic datasets.

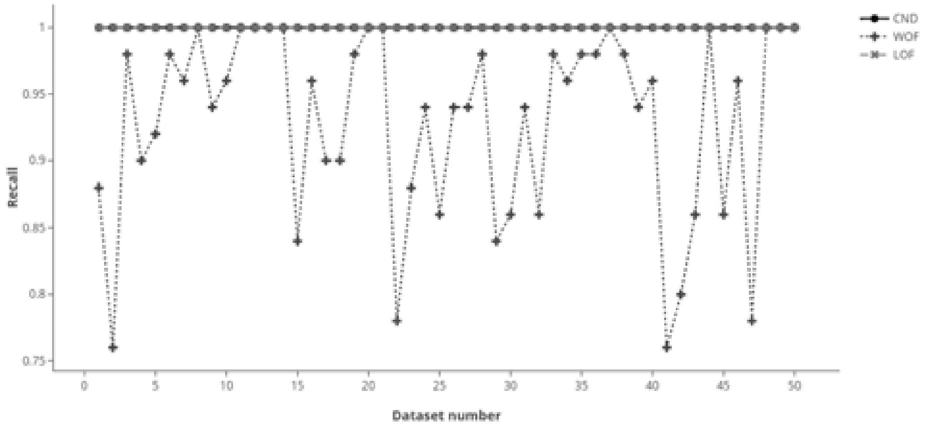


Figure 4.12**:** The recall on collection 1 of synthetic datasets.

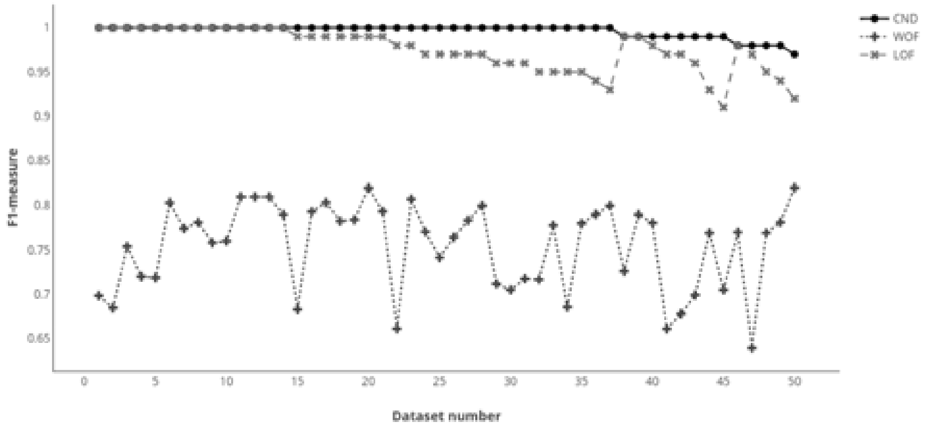


Figure 4.13**:** The F_1-measure on collection 1 of synthetic datasets.

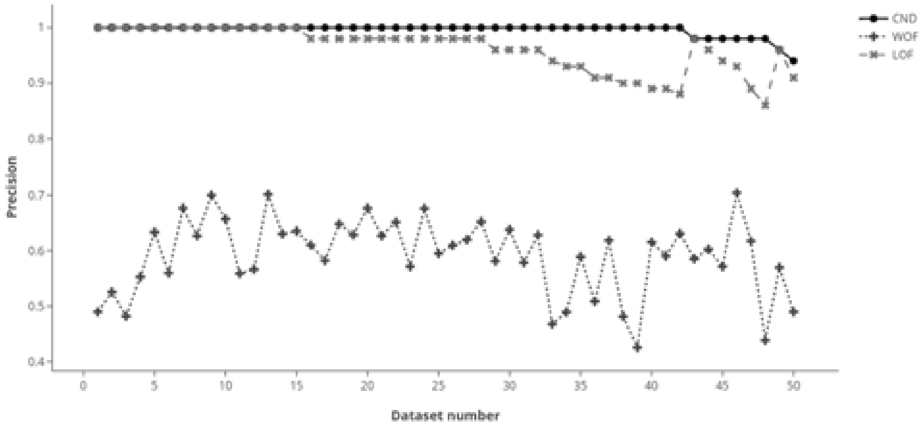


Figure 4.14**:** The precision on collection 2 of synthetic datasets.

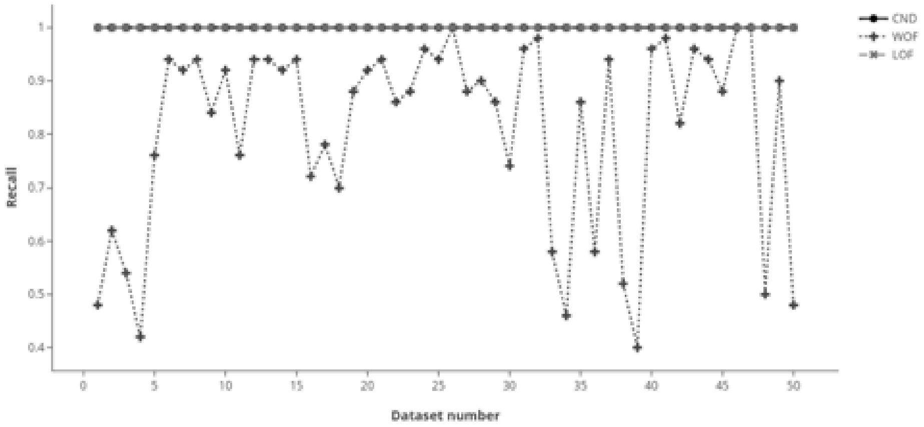


Figure 4.15**:** The recall on collection 2 of synthetic datasets.

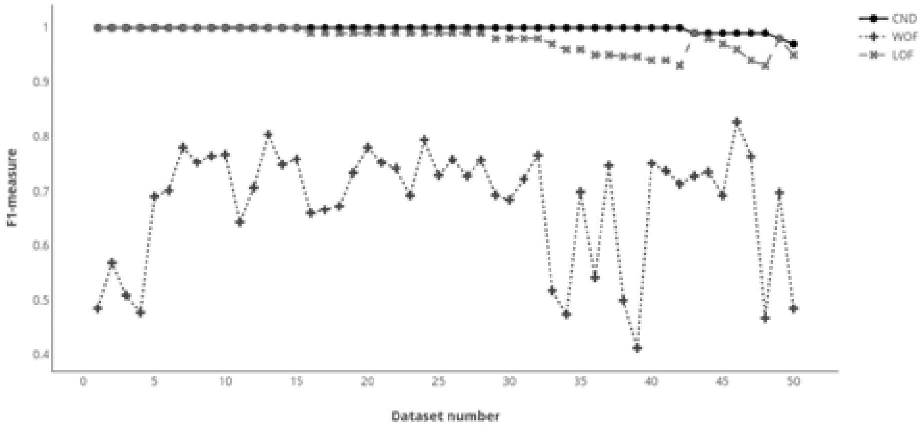


Figure 4.16**:** The F_1-measure on collection 2 of synthetic datasets.

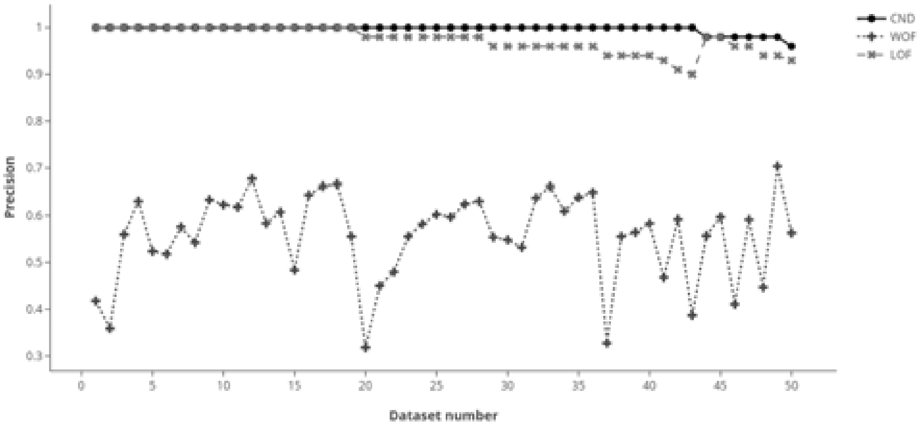


Figure 4.17**:** The precision on collection 3 of synthetic datasets.

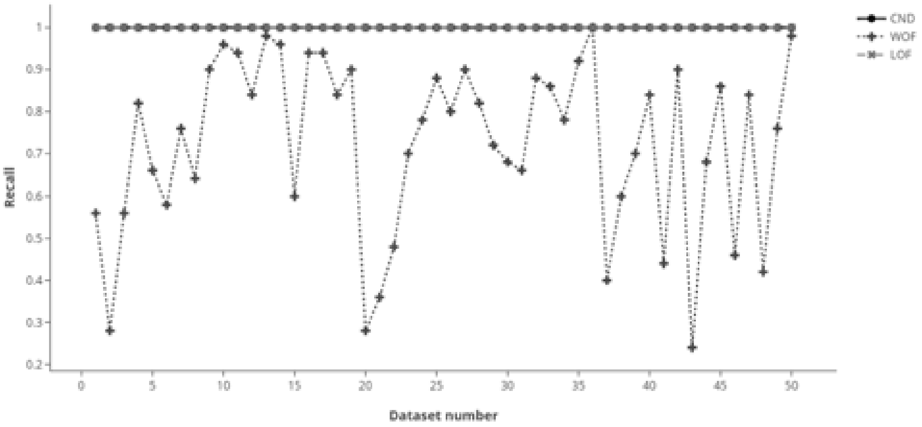


Figure 4.18**:** The recall on collection 3 of synthetic datasets.

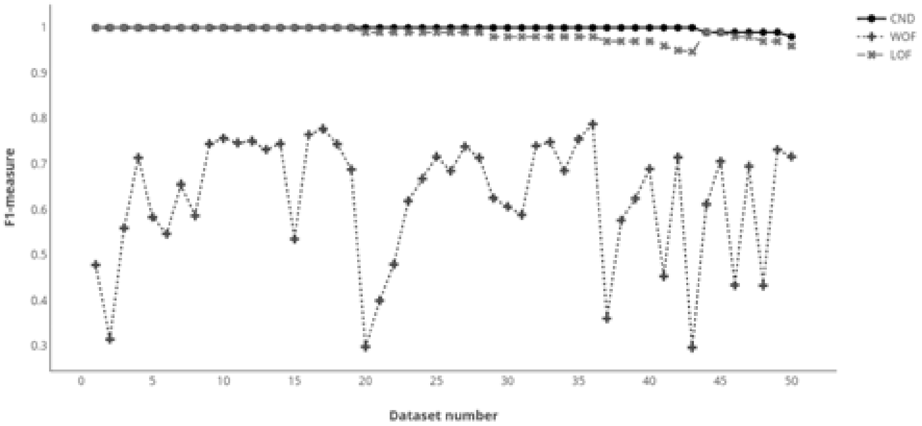


Figure 4.19**:** The F_1-measure on collection 3 of synthetic datasets.

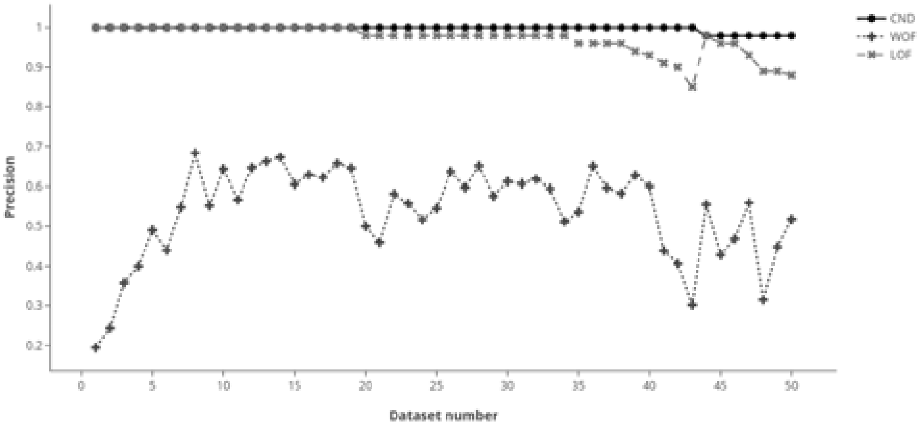


Figure 4.20**:** The precision on collection 4 of synthetic datasets.

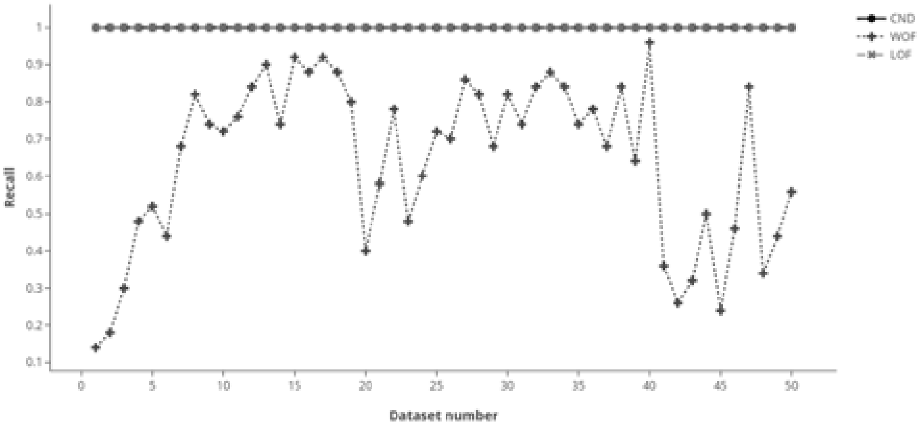


Figure 4.21**:** The recall on collection 4 of synthetic datasets.

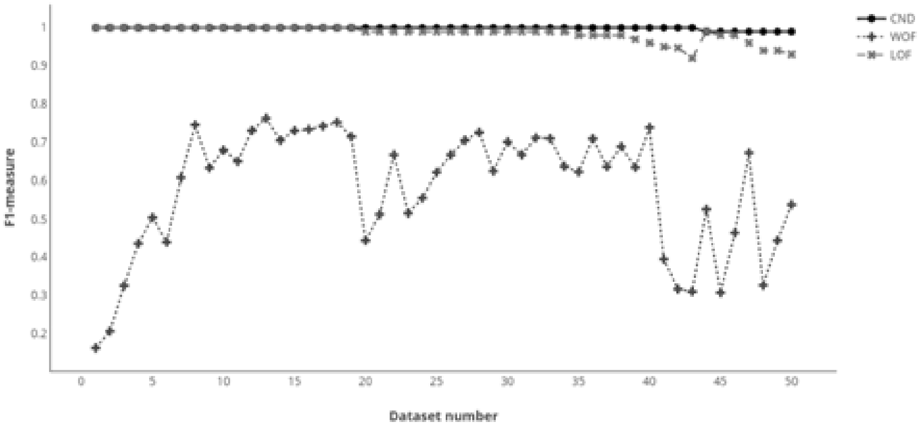


Figure 4.22**:** The F_1-measure on collection 4 of synthetic datasets.

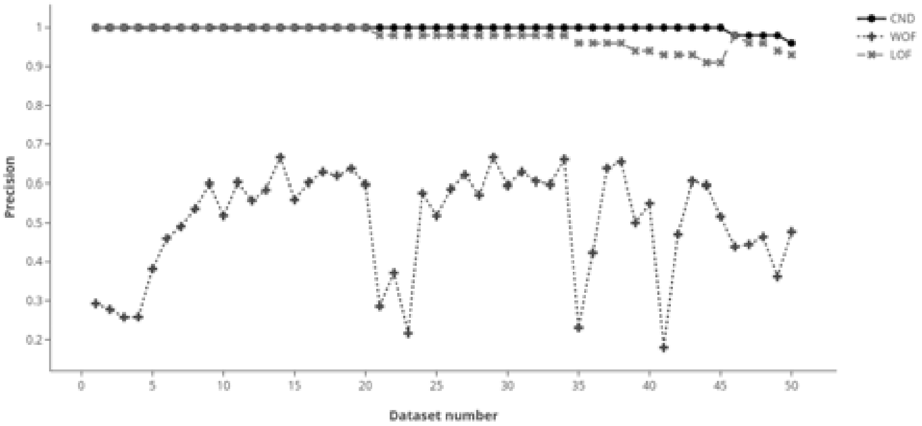


Figure 4.23**:** The precision on collection 5 of synthetic datasets.

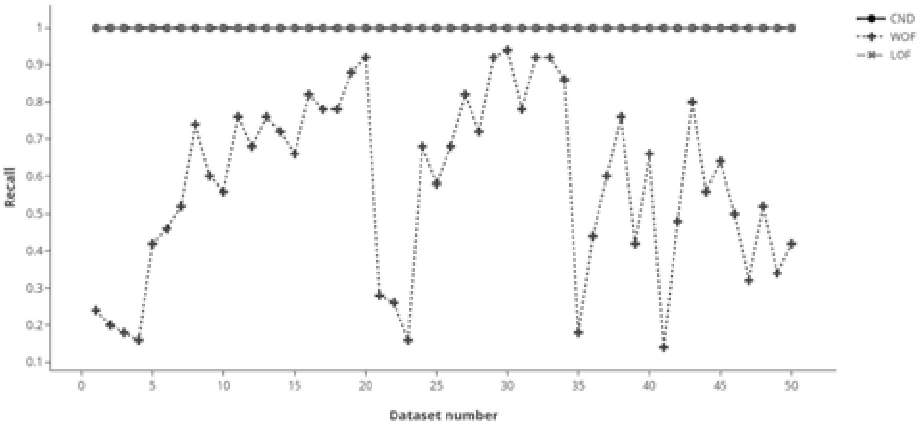


Figure 4.24**:** The recall on collection 5 of synthetic datasets.

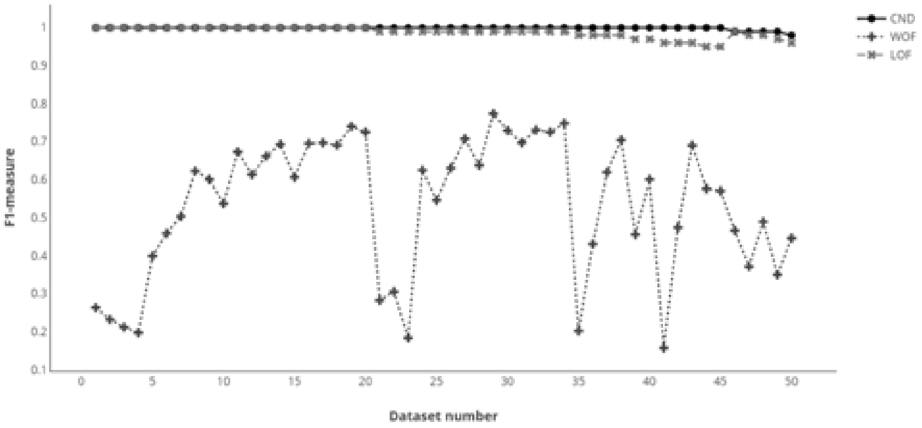


Figure 4.25**:** The F_1-measure on collection 5 of synthetic datasets.

4.2 Real-world dataset

Three real-world datasets from UCI-Machine Learning website [21] are used to test the performance of the algorithms (Iris Plants Database, Wine recognition data, and Wisconsin Diagnostic Breast Cancer). Each dataset will be divided into two class which are the outlier class (5% outliers) and the normal class to evaluate the performance of the algorithm for detecting the outliers. The detail of each dataset is shown as follows.

Iris Plants Database (IRIS)

Iris Plants Database [22] contains the sepal and petal measurements of three iris plants having four continuous-valued attributes with 150 instances (Iris Setosa (50), Iris Versicolor (50), and Iris Virginica (50)). This experiment generates ten datasets where each dataset keeps all Iris Versicolor and Iris Virginica as the normal instances and randomly picking 5 neighbor instances from Iris Setosa as an anomalous assemblage (about 5% outliers). See Figure 4.26.

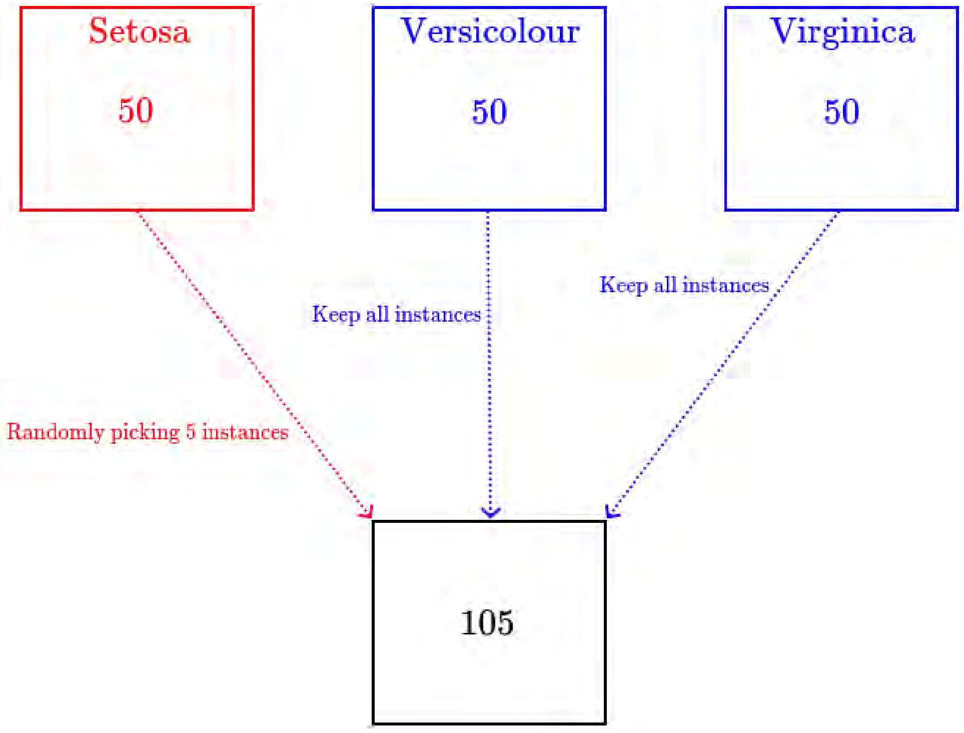


Figure 4.26: The model to generate IRIS datasets.

Wine recognition data (WINE)

Wine recognition data [23] is a dataset of the chemical analysis of wines in Italy from three different cultivars which contains thirteen continuous-valued attributes (three classes having 178 instances: class 1 (59), class 2 (71), and class 3 (48)). This experiment generates ten datasets where each dataset keeps all class 2 and class 3 as the normal instances and randomly picking 6 neighbor instances from class 1 as an anomalous assemblage (about 5% outliers), see Figure 4.27.

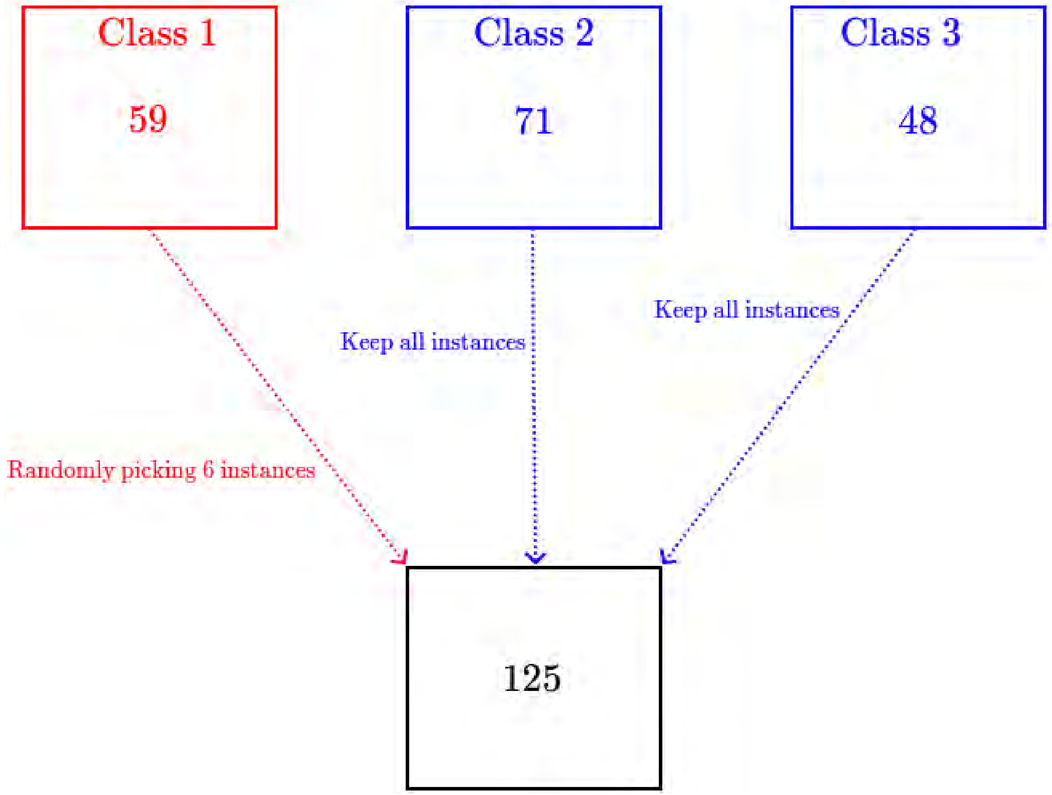


Figure 4.27**:** The model to generate WINE datasets.

Wisconsin Diagnostic Breast Cancer (WDBC)

Wisconsin Diagnostic Breast Cancer [24] is a dataset of the characteristics of cell nucleus in the diagnosis breast cancer image which contains thirty one attributes (ID, and 30 continuous-valued attributes) with two classes having 569 instances (malignant (212), and benign (357)). This experiment drops ID attribute and generates ten datasets where each dataset keeps all benigns as the normal instances and randomly picking 18 neighbor instances from malignant as an anomalous assemblage (about 5% outliers), see Figure 4.28.

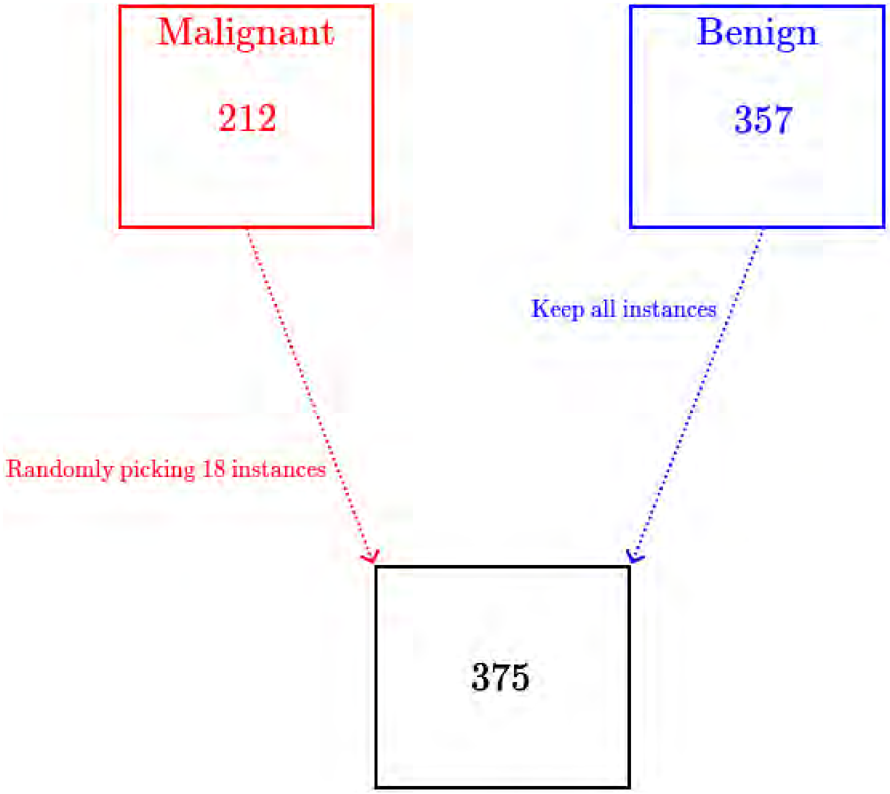


Figure 4.28: The model to generate WDBC datasets.

Real-world datasets	**A normal cluster**	**An anomalous assemblage**
IRIS	100 instances (versicolor (50), virginica (50))	5 neighbor instances (randomly from setosa (50))
WINE	119 instances (class 2 (71), class 3 (48))	6 neighbor instances (randomly from class 1 (59))
WDBC	357 instances (benign (357))	18 neighbor instances (randomly from malignant (212))

Table 4.2: The summary of the randomly generated data from real-world datasets.

To determine the suitable value of the parameter C for real-world datasets. CND is performed by vary the parameter $C = 1$ to 10 on each dataset in IRIS, WINE, and WDBC where F_1-measure of each dataset is computed. Next, the average $F1$-measure (AF) of ten generated datasets in IRIS, WINE, and WDBC is computed and the standard deviation of this average $F1$-measure (SDAF) is also computed. Finally, the error bar is plotted by AF ± SDAF of ten generated datasets in IRIS, WINE, and WDBC along the value of $C = 1$ to 10, see in Figure 4.29.

Discussion of error bars on real-world datasets

For IRIS: $C = 6$ shows the best average F_1-measure about 1.0 and a small standard deviation. After that, the average F_1-measure is stable about 1.0 from $C = 7$ to 9. Moreover, the average F_1-measure drops to 0.1 at $C = 10$ because the scores of instances may not be significantly different or there are some normal instances are detected as outliers. For WINE: $C = 7$ shows the best average F_1-measure about 0.8. After that, the average F_1-measure is stable about 0.7 from $C = 8$ to 10. For WDBC; $C = 7$ shows the best average F_1-measure about 0.6. After that, the average F_1-measure is stable about 0.6 from $C = 8$ to 10.

Remark. $C = 7$ guarantees the best performance for these three real-world datasets.

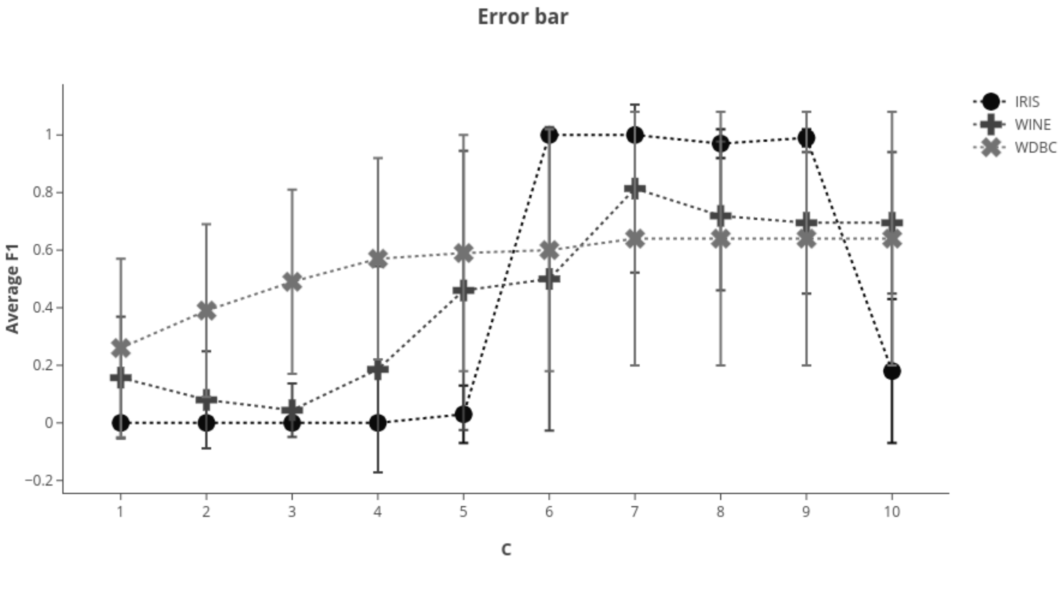


Figure 4.29**:** The error bars of IRIS, WINE, and WDBC where C along x-axis and the average F_1-measure along y-axis.

To compare the performance of CND with WOF and LOF on real-world datasets, the parameter C is set to 7 for CND and $k = \left\lfloor \frac{7}{100} \times m \right\rfloor$ for LOF. The results are shown in Figure 4.30 - 4.38.

Discussion of the performance on real-world datasets

On IRIS datasets: CND shows the best performance with 100% for all measurements which is better than WOF and LOF.

On WINE datasets: CND is better than WOF and LOF on the dataset number 1,2,3,4,7,8,9 for all measurements, CND has similar performance to LOF which is better than WOF on the dataset number 5 and 6 for all measurements, and CND has similar performance to LOF which is worse than WOF on the dataset number 10 for all measurements.

On WDBC datasets: CND is better than WOF and LOF on the dataset number 1 - 7 for all measurements, CND has similar performance to WOF which is worse than LOF on the dataset number 8 for all measurements, and these three algorithms have the same performance on the dataset number 9 and 10.

For overall performance on these three real-world datasets, the results show that CND is better than WOF and LOF based on precision, recall, and F_1-measure. Since the outlier class may be close to the normal class, then these algorithms can not effectively detect the outliers on these three real-world datasets. Moreover, the threshold based on the adjusted boxplot may be unsuitable for WOF and LOF as shown in three real-world datasets.

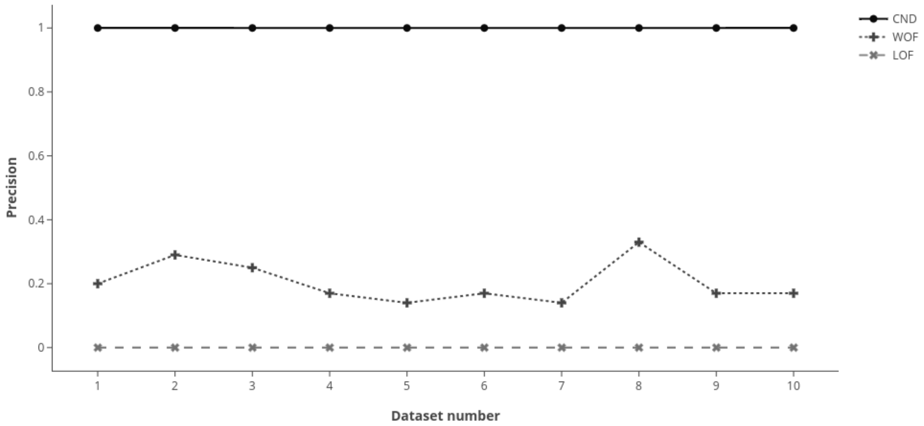


Figure 4.30**:** The precision on IRIS datasets.

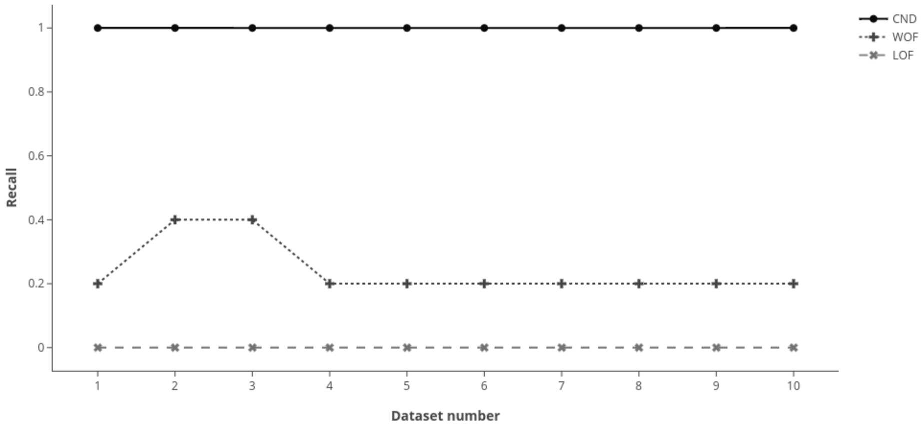


Figure 4.31**:** The recall on IRIS datasets.

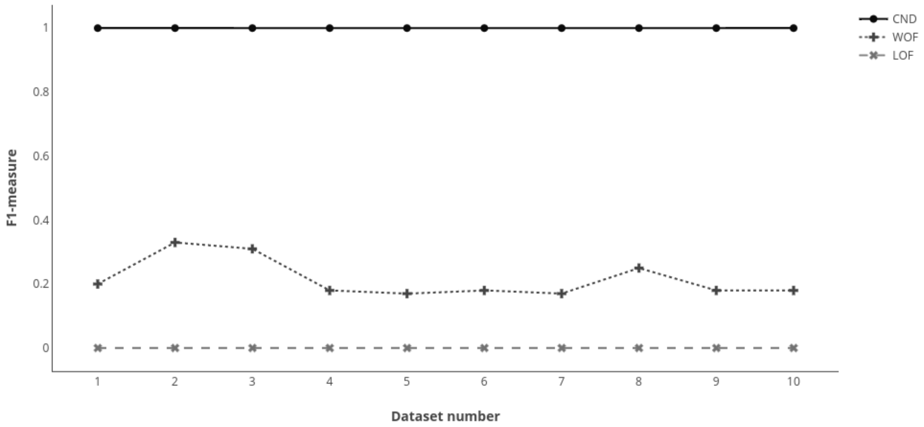


Figure 4.32**:** The F_1-measure on IRIS datasets.

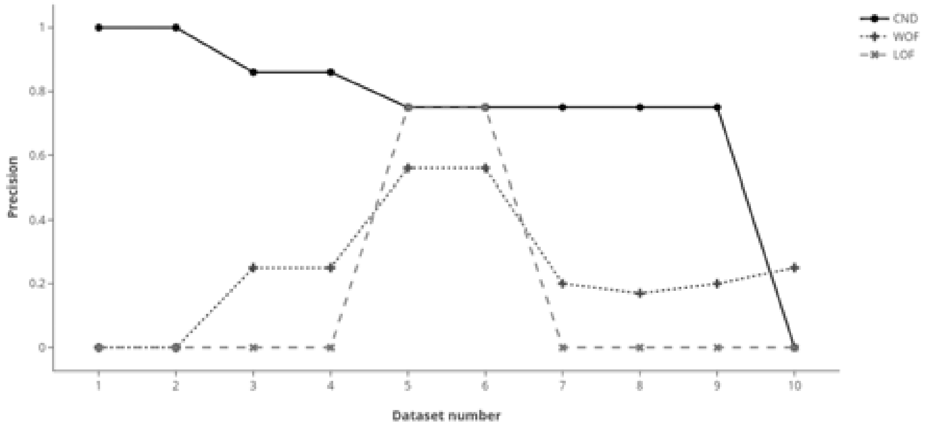


Figure 4.33**:** The precision on WINE datasets.

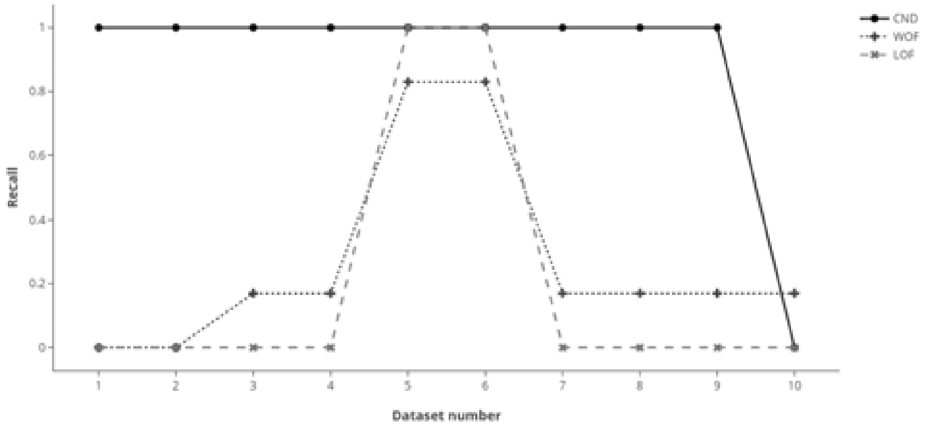


Figure 4.34**:** The recall on WINE datasets.

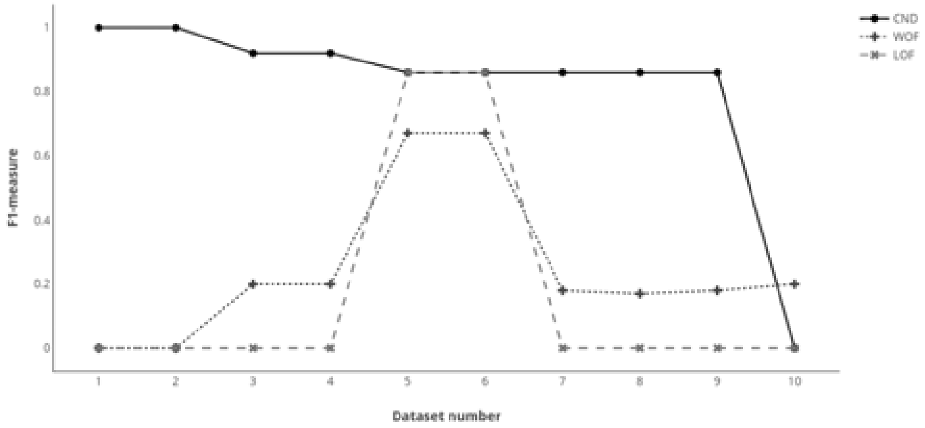


Figure 4.35**:** The F_1-measure on WINE datasets.

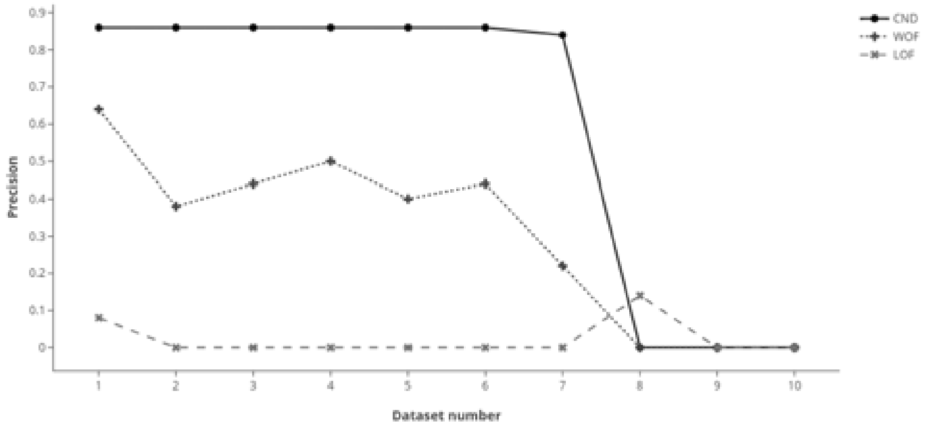


Figure 4.36**:** The precision on WDBC datasets.

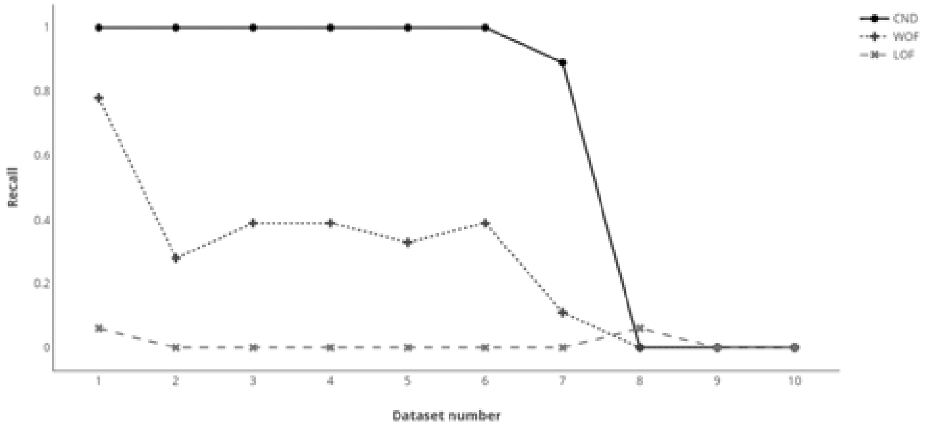


Figure 4.37**:** The recall on WDBC datasets.

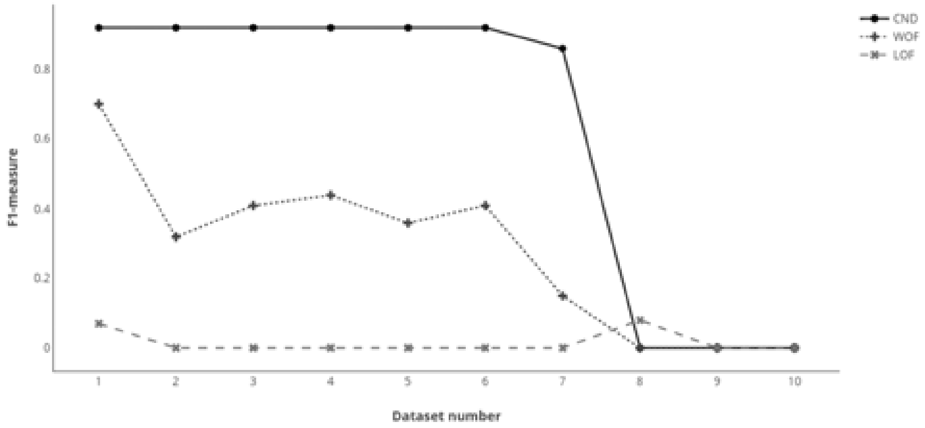


Figure 4.38**:** The F_1-measure on WDBC datasets.

4.3 Running time

Generated datasets vary by the number of instances from 100 to 1000 instances to test the running time of CND, WOF, and LOF. This experiment is performed by Julia programming language version 0.5 on a personal computer notebook (Intel Core i7, 8GB memory). The result is shown in Figure 4.39 where CND has the running time is similar to WOF which is better than LOF.

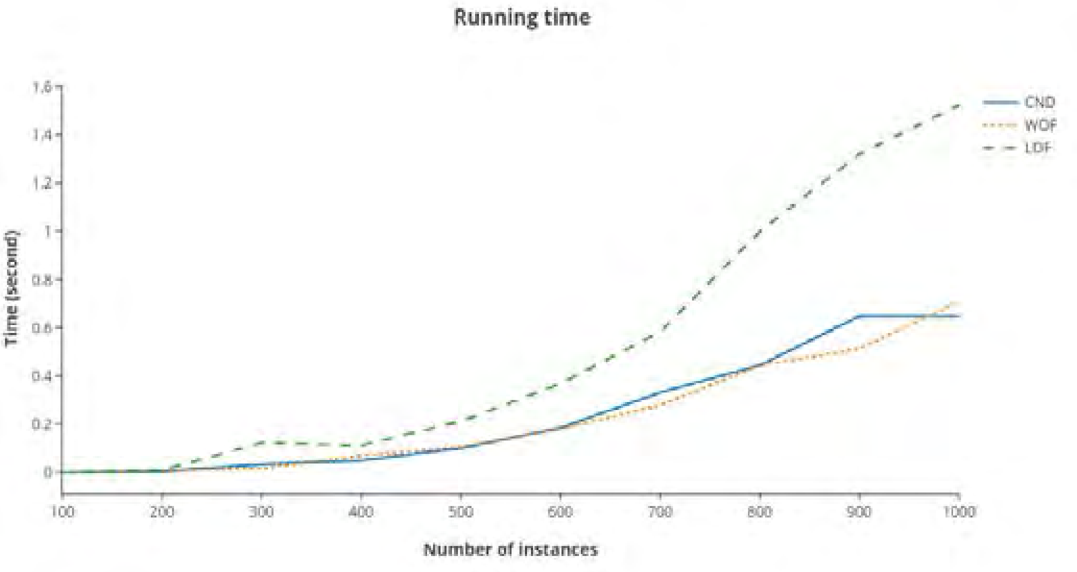


Figure 4.39**:** The running time of CND, WOF, and LOF.

CHAPTER V

CONCLUSION

In this thesis, the C-anomalous assemblage detection algorithm called CND is proposed. The algorithm needs the parameter C to compute the k^{th} index to use the k^{th}-nearest neighbor distance for representing the anomalous score of each instance in a dataset. The index k is set to equal the floor function of C percent of the total number of instances in a dataset. Moreover, the adjusted boxplot based on the medcouple for skew distribution is used to generate the threshold for detecting outliers where CND algorithm uses only the upper threshold because any outlier has a large score.

The effect of varying parameter C is investigated by the error bars of average F_1-measure along the value of C from 1 to 10 on synthetic and real-world datasets having about 5% outliers. For synthetic datasets where the anomalous assemblages are deliberately designed, the lesson that has been learned is the suitable value of parameter C depends on the size of the anomalous assemblages. Moreover, the various locations of anomalous assemblages around the normal clusters have no effect because CND shows the good performance for all different locations of assemblages in the experiment. For real-world datasets, the results are not clear to identify the suitable value of parameter C because there are no obvious anomalous assemblages appearing in them.

On the experimental results, CND shows the best prediction and effective score instances based on precision, recall, and F_1-measure on five collections of synthetic and three real-world datasets (IRIS, WINE, and WDBC). Moreover, the time complexity of CND is $O(n^2)$ which is the same as WOF and LOF. However, the running time of CND is similar to WOF which is better than LOF.

Future work

The performance of CND should be evaluated via more well-designed on the non-normal distributed datasets. See the example in Figure 5.1.

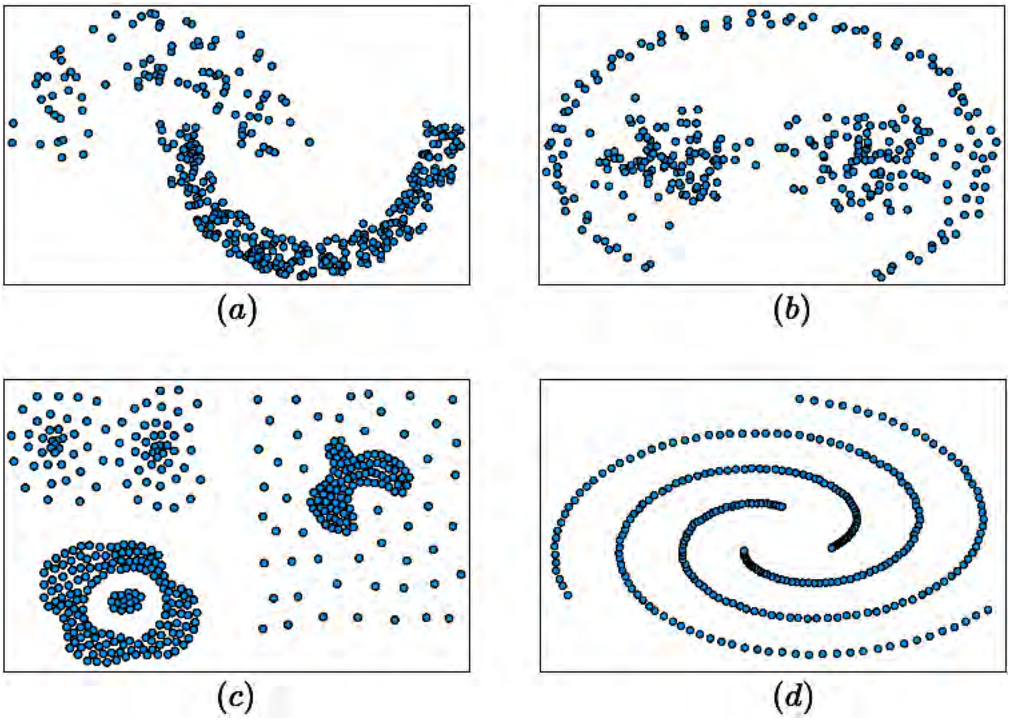


Figure 5.1: Non-normal distribution datasets ([25], [26], [27]).

On a dataset in Figure 5.1-(a),(b), and (d), CND may identify all instances as the normal instances because each instance has the distance to its neighbors that do not significantly different comparing with the most instances.

On a dataset in 5.1-(c), CND may identify some instances are the outliers because there are the instances having the distance to the neighbors that farther away from comparing with the most instances.

REFERENCES

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

[2] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: DaWaK, Vol. 2454, Springer, 2002, pp. 170–180.

[3] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial intelligence review 22 (2) (2004) 85–126.

[4] C. C. Aggarwal, An introduction to outlier analysis, in: Outlier analysis, Springer, 2013, pp. 1–40.

[5] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 15.

[6] C. C. Aggarwal, P. S. Yu, Outlier detection for high dimensional data, in: ACM Sigmod Record, Vol. 30, ACM, 2001, pp. 37–46.

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

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

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

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

[11] J. Tukey, W.(1977). exploratory data analysis, Reading: Addison-Wesley.

[12] M. Hubert, E. Vandervieren, An adjusted boxplot for skewed distributions, Computational statistics & data analysis 52 (12) (2008) 5186–5201.

[13] P. Zezula, G. Amato, V. Dohnal, M. Batko, Similarity search: the metric space approach, Vol. 32, Springer Science & Business Media, 2006.

[14] V. Perlibakas, Distance measures for pca-based face recognition, Pattern Recognition Letters 25 (6) (2004) 711–724.

[15] D. T. Larose, K-nearest neighbor algorithm, Discovering Knowledge in Data: An Introduction to Data Mining (2005) 90–106.

[16] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE transactions on information theory 13 (1) (1967) 21–27.

[17] I. Tajuddin, On the use of medcouple as a measure of skewness., Pakistan Journal of Statistics 28 (1).

[18] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (8) (2006) 861–874.

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

[20] G. E. Batista, R. C. Prati, M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 20–29.

[21] A. Asuncion, D. Newman, Uci machine learning repository (2007).

[22] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of human genetics 7 (2) (1936) 179–188.

[23] S. Aeberhard, D. Coomans, O. De Vel, Comparison of classifiers in high dimensional settings, Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep (92-02).

[24] W. N. Street, W. H. Wolberg, O. L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis.

[25] A. K. Jain, M. H. Law, Data clustering: A user's dilemma, PReMI 3776 (2005) 1–10.

[26] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, Pattern Recognition 41 (1) (2008) 191–203.

[27] C. T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Transactions on computers 100 (1) (1971) 68–86.

BIOGRAPHY

Name	Mr. Kayyasit Singkarn
Date of Birth	21 April 1993
Place of Birth	Yasothon, Thailand.
Education	Bachelor of Science (Mathematics) , Chulalongkorn University, 2014.

Publication

- K. Singkarn, and K. Sinapiromsaran, "C-anomalous assemblage detection to recognize outliers using k^{th}-nearest neighbor distance" in Computer Science and Engineering Conference (ICSEC), 2017 International, IEEE, pp. 225-230.