Doctoral Dissertation

Outlier Detection and Class Imbalance Based on Mass Estimation Integrated with Evidential Reasoning

HOANG, Anh

Supervisor: Professor HUYNH, Van-Nam

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Knowledge Science)

September 2021

Abstract

Outlier detection and class imbalance modeling process play significant roles to enable effective and efficient algorithms for statistic analysis, data mining, machine learning, and knowledge discovery frameworks working on imbalanced datasets. Although there has been vast literature on imbalanced datasets, the shortcomings of distance-based functions in response to a varied density of data points have not been solving yet.

The primary aim of this dissertation was to exploit a new alternative approach for local outlier detection tasks by fundamentally changing the way to measure the outlier degree of each data point. To achieve this goal, we developed a mass-based approach to measure the dissimilarity between data points. Then, we introduced a new outlier scoring method by employing mass-based dissimilarity and probability modeling to detect the local outliers in a given dataset. The experimental study tested on artificial datasets and real application datasets show that our proposed MLOS approach is competitive with the state-of-the-art approaches.

In the same manner, to exploit the mass-based measurement for learning from the imbalanced datasets, we introduce the other two new methods for the class imbalance task. The first model is a simple application of weighted sum. The second model is an integration of the mass estimation and the Dempster-Shafer theory of evidence. These proposed models were assessed by using significant evaluation metrics such as F1 score, Brier score, ROC-AUC, and PR-AUC score testing on a wide range of benchmark datasets. In addition, all experimental results were validated using the non-parametric statistical Wilcoxon signed ranks test.

This dissertation was the first study, regarding to our knowledge, to investigate the local outlier detection problem using mass-based dissimilarity measurement; the key finding was that the proposed MLOS approach presents an alternative way to score the outlierness of each data point in a given dataset. Secondly, the simulation results showed that our proposed new models for the class imbalance task outperformed the other 11 competitive methods. The experiments were conducted on a wide varying application domains, a varied imbalance ratio, and the number of instances.

Keywords: Imbalanced data, outlier detection, outlier modeling, mass-based dissimilarity, weighted sum, Dempster-Shafer theory.

Acknowledgment

Writing the acknowledgment is always the nicest part! First, I would like to mention the International Cooperation Department, Ministry of Education and Training, Viet Nam, for providing me a scholarship as a part of Project 911. I could not start my Doctor of Philosophy program at Japan Advanced Institute of Science and Technology (JAIST) without this financial support.

Special thanks to my supervisor, Professor HUYNH Van-Nam, for all he has done, which I will never forget. I truly appreciate his time spending to help me on many occasions with exceptional supports. Thanks for sharing knowledge not only by doing scientific research but also by living a happy life. Besides, I received generous encouragement and assistant from the HUYNH's Lab. members in this work, especially Mr. Toan and Mr. Vinh.

I would like to express my thankfulness to Professor HASHIMOTO Takashi, for his wonderful course of Introduction to Knowledge Science (K218). I enjoyed every minute of the lectures as well as the discussion in the official hours.

Professor DAM Hieu-Chi, thank you so much for caring about both what and how you have been teaching us. I have learned a lot of fundamental concepts and methodologies for doing data scientist from your courses. In addition, I would particularly like to mention enjoying time for playing soccer together.

My sincere thanks go to JAIST Supercomputer Unit for running software and services smoothly to conduct the experimental studies. Thanks to Student Welfare Section for supporting my living at JAIST. Thanks to Educational Service Section, Secretarial Service Section, and other sections at JAIST for unconditional help.

Last and most of all, I am grateful to the committee members and the audiences, who might give me the questions and comments. That will help a lot to improve my work.

Finally, a lot of people have supported me, and I relish this opportunity to thank them. Thanks to the members of JAIST's Football Club, who may leave everything behind and enjoy doing sport together. Thanks to my colleagues and friends who often ask me about my health and my progresses. Especially, my parents are always believing in me. My spouse and my son, thank you so much for being part of my life.

This research was supported in part by the US Office of Naval Research Global under grant no. N62909-19-1-2031.

Contents

Abstract **II**

Acknowledgment **III**

Contents **IV**

List of Figures **VII**

List of Tables **VIII**

List of Abbreviations **IX**

Chapter 1 Introduction **1**
1.1 Research problems 1
1.1.1 Outlier detection 2
1.1.2 Class imbalance 3
1.2 Research questions and contributions 5
1.2.1 Research questions 5
1.2.2 Main contributions 5
1.2.3 Future directions 6
1.3 Dissertation organization 7

Chapter 2 Research background **8**
2.1 Hierarchical partitioning method 8
2.2 Mass-based dissimilarity measurement 8
2.2.1 Definition 1 9
2.2.2 Definition 2 10
2.2.3 k-lowest mass-based dissimilarity neighbors 11
2.3 Dempster-Shafer theory 11
2.4 Evaluation metrics 12
2.5 Non-parametric statistical analysis 14

Chapter 3 Outlier detection **15**
3.1 Introduction 15

3.2 Problem formulation 18
3.3 Literature review 18
3.3.1 Geometric outlier modeling 18
3.3.2 Semi-supervised outlier modeling 20
3.4 Proposed MLOS approach 21
3.4.1 Notations 23
3.4.2 Stage 1: Data preparation 23
3.4.3 Stage 2: Data partitioning technique 23
3.4.4 Stage 3: Outlier scoring 24
3.5 Experimental result 28
3.5.1 Experimental results on synthetic datasets 28
3.5.2 Experimental results on benchmark datasets 34
3.5.3 Non-parametric statistic test 44
3.6 Chapter conclusions 46

Chapter 4 Class imbalance **48**
4.1 Introduction 48
4.2 Class imbalance statement 50
4.3 Methodology 51
4.3.1 Confidence estimation 51
4.3.2 Mass-based similarity measurement 51
4.3.3 Mass-based similarity weighted k-neighbor Sk-LMN approach 52
4.3.4 Mass-based similarity integrated with evidential reasoning: EMass approach 55
4.4 Experimental studies 58
4.4.1 Dataset description 58
4.4.2 Implementation details and evaluation metrics 62
4.4.3 Results and discussions 63
4.5 Chapter conclusions 69

Chapter 5 Conclusions and future works **71**

Publications **73**

References **74**

Appendix A **84**

Appendix B **89**

List of Figures

2.1 Isolation Forest (*Source* : *towardsdatascience.com*) 9
2.2 Results of calculated mass-based dissimilarity. 10

3.1 reach-dist(x_1, y) and reach-dist(x_2, y) for $k = 4$. 19
3.2 The proposed MLOS approach for detecting local outliers. . . 22
3.3 Results tested by MLOS approach. 29
3.4 Results tested by LOF approach. 29
3.5 Results tested by IForest approach. 30
3.6 Results tested by LOOP approach. 31
3.7 Outlier detectors comparison tested on five 2D datasets. . . . 33
3.8 ROC curves tested on the WBC dataset. 42
3.9 PR curves tested on the WBC dataset. 43
3.10 ROC curves tested on the Hepatitis dataset. 43
3.11 PR curves tested on the Hepatitis dataset. 44

4.1 Flowchart of the Sk-LMN approach. 53
4.2 Flowchart of the EMass approach. 56
4.3 Sk-LMN comparison results on PR-AUC results. 64
4.4 Sk-LMN comparison results on F1 scores. 64
4.5 EMass comparison results on F1_score. 66
4.6 EMass comparison results on Brier_score. 67
4.7 EMass comparison results on ROC-AUC. 67
4.8 EMass comparison results on PR-AUC. 68

5.1 ROC curves tested on the Glass dataset. 84
5.2 PR curves tested on the Glass dataset. 85
5.3 ROC curves tested on the Cardio dataset. 85
5.4 PR curves tested on the Cardio dataset. 86
5.5 ROC curves tested on the Shuttle dataset. 86
5.6 PR curves tested on the Shuttle dataset. 87
5.7 ROC curves tested on the Parkinsons dataset. 87
5.8 PR curves tested on the Parkinsons dataset. 88

List of Tables

2.1 Confusion matrix 13

3.1 The property of 25 real-application datasets for detecting outliers. 34
3.2 The average accuracy results tested on 25 real application datasets. 39
3.3 The comparison of F1 score tested on real application datasets. 40
3.4 The comparison of AUC results tested on real application datasets. 42
3.5 Wilcoxon signed ranks test for accuracy metric. 45
3.6 Wilcoxon signed ranks test for F1 metric. 45
3.7 Wilcoxon signed ranks test for ROC-AUC metric. 46
3.8 Wilcoxon signed ranks test for PR-AUC metric. 46

4.1 Descriptions of 60 imbalanced datasets. Idx., #Inst., #Ftr., and IR represent index of dataset, number of instances, features, and imbalance rate respectively. 59
4.2 Wilcoxon signed ranks test on Sk-LMN comparison results. . . 65
4.3 Wilcoxon signed ranks test on EMass comparison results. . . . 69

5.1 F1 score results comparisons. 89
5.2 Brier score results comparisons. 90
5.3 ROC-AUC results comparison. 91
5.4 PR-AUC results comparison. 92

List of Abbreviations

Abbreviations	**Terms**
AUC	Area Under the Curve
ANOVA	Analysis of Variance
BPA	Basic Probability Assignment
DT	Decision Tree
DST	Dempster-Shafer Theory
IForest	Isolation Forest
ITree	Isolation Tree
k-NN	k-Nearest Neighbors
k-LMN	k Lowest Mass Neighbors
KDD	Knowledge Discovery in Databases
KEEL	Knowledge Extraction based on Evolutionary Learning
LR	Logistic Regression
LinearSVM	Linear Suport Vector Machine
LOF	Local Outlier Factor
LOOP	Local Outlier Probability
RF	Random Forest
MLOF	Mass-based Local Outlier Factor
MLOS	Mass-based Local Outlier Score
NB	Naive Bayes
ODDS	Outlier Detection Datasets
One-SVM	One-Class Support Vector Machine
PR	Precision-Recall
ROC	Receiver Operating Characteristic Curve
SMOTE	Synthetic Minority Oversampling Technique

Chapter 1

Introduction

1.1 Research problems

Making a binary decision is one of the most frequent human cognitive activities, for instance, an alternative between taking a specific action or not taking it. As a matter of fact, binary decisions are essential to many other fields. Statistical analysis, data mining, machine learning, and knowledge discovery are not the exception. In those fields, tasks such as outlier modeling and class imbalance involve learning from the imbalanced datasets to support decision making. Motivated by a wide range of application domains and an interesting scientific research topic under viewpoints of a binary decision problem, this dissertation investigates the outlier detection and class imbalance tasks. Although there has been vast literature on these tasks, almost all algorithms based on distance functions show the shortcomings. Therefore, this work aims to learn new models, which can handle the key weaknesses of the distance-based functions, from the imbalanced dataset for making a rational binary decisions.

This dissertation is also motivated by three main concepts: binary decision making, binary questions and answers, and decision rules or models. These core concepts fit to three stages in a regular data analysis, which are represented in the research questions.

- Conceptual formulation: For outlier detection task, in a given dataset, whether a considering instance is being an outlier or not; How to score the outlierness of an instance. For class imbalance task, which label (minority or majority) does a query instance belong to.
- Analysis task: how a novel approach is developed or learnt from the given dataset.
- Conclusion: when to make a binary decision, such as outlier/inlier or class label.

1.1.1 Outlier detection

Outlier is defined as an unusual instance that differs significantly from the remaining parts of a given dataset. Outlier detection is an influential aspect of statistical analysis, data mining, machine learning, and knowledge discovery frameworks to identify emerging and delightful patterns, anomalies, and trends from a given dataset. The outlier modeling task might be studied as a binary classification task challenging on the imbalanced dataset. Because each instance is determined as an outlier or an inlier. This field has been widely studying in statistics, data mining, and machine learning [1] with various names such as anomaly detection, deviation detection, novelty detection, exception mining, and outlier mining.

Summarily, an outlier detection solution intends to figure out an unknown truth from the observations whose inferential target is a binary truth such as inlier or outlier. For example, deciding whether an instance being a local outlier in a given dataset is an inferential question whose answer is unknown. Therefore, the outlier detector is constructed consequent to the concept of statistical significance. That intuitively means the observations present strong evidence to infer the outliers.

Researchers have obtained remarkable successes in a varied applications as briefly reviewing here:

- Anomaly-based network intrusion detection: [2] Network intrusion detection systems (NID) make decision on whether network traffic is normal (good) or abnormal (bad) automatically. These systems use machine learning solutions such as classification and clustering methods to distinguish abnormal traffic from normal ones. In other words, that is a comparison between the model of anomalous traffic and the model of normal ones. These distinguish often rely on the ability to assess the dissimilarity or distance between a target object and a labeled one to determine whether a query target anomalous or not.
- Anomaly detection techniques in the finance domain: [3] Abnormal behavior in a credit card transaction could represent fraudulent activities. There have been many works on clustering-based approach for fraudulent detection, which is a critical requirement to protect the customers. New fraud transactions have been inventing nowadays. However, the existing techniques may unable to detect them. Hence, it is significant to develop new efficient and effective algorithms to determine unusual transactions.
- Outlier detection in urban traffic data: [4] The analysis of urban traffic aims to learn from the behavior of participants. These participants include cars, trucks, public transportation. City managers consider

different situations on weather, events, street conditions to make decisions. Therefore, outlier detection on traffic flow data and the relation of outliers to specific circumstances plays an important role in traffic management for the urban areas.

- Social media anomaly detection: [5, 6] To prevent malicious activities such as bullying, terrorist attack, and fraud information dissemination, the applications of anomaly detection is critically important with the recent popularity of social media.
- Detecting the signal of a machine failure by applying anomaly detection methods: [7] The application of anomaly detector is developed for early warning of faults for condition-based maintenance because of the limited lifetime of any machines or components. Therefore, monitoring the condition of machines and particulars, maintaining a desirable working state becomes a crucial task.
- Video surveillance: [8] Surveillance systems have been widely used due to the increasing demand for security. Abnormal event detection is a primary challenge of video analysis that identifies unexpected events or patterns. The video of normal events is utilizing as training dataset for learning new model. Then, to detect the abnormal events that might not confirm to the learned model.

1.1.2 Class imbalance

Equally important to the outlier detection, the purpose of a classification task is to predict an unknown property of a query instance, such as which class the new instance belongs to, based on the available attributes of this instance. Binary classification solution is a type of prediction approach. That target property is binary decision. In machine learning field, the binary classification belongs to supervised learning since the model is learnt from training dataset with a set of feature values.

In particular, we focus on classifying the imbalanced dataset where the class distribution of the data points is skewed. In other words, the imbalanced classification task involve the number of instances representing one class is much lower than the ones of the other classes. This problem is known to interrupt the performance of classifying models due to their accuracy-oriented evaluation metric, which often makes the minority class be forgotten.

We first learn a classifier from the training data, then apply the learned model to predict unknown labels of unseen instances from their observed feature values. This investigation aims to introduce algorithms that automatically learn model or prediction rules from training data. From literature review, there is a wide range of application domains where the class imbalance

problem is presented in the real world as follows:

- Engineering: The authors of [9] aim to detect oil spills in satellite radar images by using specific approaches to handle data imbalance. The other example is to detect defects or faults in semiconductors that have been addressed in [10,11]. Among others, in [12] the authors dealt with wind turbine failure prediction. This prediction task is essential for operating a wind farm due to its maintenance costs. To reduce these costs, the best design is being able to automatically monitor, diagnose and predict the state of wind turbines.
- Bioinformatics: Protein datasets are always imbalanced, then a specific technique is required to identify the protein structures and functions. This requirement is a significant problem in the bioinformatics and biotechnology domain due to the functioning of an organism. The authors of [13] focused on protein data classification. Besides, cell recognition is also an application of class imbalance techniques. For example, the detection of micro ribonucleic acid (RNA) is introduced in [14] or the mitotic cells could be detected in HEp-2 images [15].
- Medical diagnosis: The authors of [16, 17] developed a computer-aided system to detect lung nodule in computer tomography images for early-stage lung cancer diagnosis. In [18], the case of breast cancer detection is considered a class imbalance task because there are many more examples from the benign class than from the malign one. Breast cancer can also be detected from magnetic resonance images (MRI) as introduced in [19].
- Business management: In [20], the authors dealt with several finance problems, such as stock market prediction, credit card approval, and fraud detection. These problems affected the accuracy and interpretability of learning models. Hence, the authors introduced a new fuzzy rule-based classification model that can deal with imbalanced data and improve prediction performance in the financial domain.
  Another crucial example is churn prediction application for customer relationship management as introduced in [21, 22]. Customer churn prediction is a class imbalance task because few customers tend to move to a different company.
- Security: Biometric authentication [23] and video surveillance are two essential security applications with imbalanced class distributions. In [24, 25], a face recognition model is introduced to detect the presence of target individuals in various scenarios. At learning time, few target label is available that lead to class imbalance problem.
- Education: Data mining techniques are applied to improve the quality

of educational services for society. In this field, the problem of detecting early school dropout [26] involves the class imbalance tasks. Training data that contains student's information at the start of the course is available to learn a model. Then, the learned model can be employed to detect whether students will withdraw the school during the course or not.

1.2 Research questions and contributions

This research tackles the drawbacks of the distance-based functions to measure the dissimilarity between instances for outlier detection and class imbalance problems, especially in the case of a dataset that has a varied density of data points. In this research, the mass-based dissimilarity measurement replaces the distance-based calculation and the remaining components are kept the same as in the conventional algorithms for both tasks. Respectively, the weighted sum and Dempster-Shafer theory of evidence are exploited for classifying imbalanced datasets.

1.2.1 Research questions

This dissertation aims to answer the following questions within the scenario that a given dataset has a varied density of data points. In this situation, the distance-based calculations in traditional statistics or conventional machine learning algorithms may have shortcomings.

- What are the conceptual formulations of the outlier and outlier detectors?
- How to construct a new model or a decision rule for detecting local outliers in a given dataset?
- What are the conceptual formulations of an imbalanced dataset and class imbalance problem?
- How to construct a new model or a decision rule for classifying the imbalanced dataset?
- How to select a good model based on what evaluation metrics? By comparing the performances with the previous studies or the baselines models.

1.2.2 Main contributions

As a result, there are main contributions from this work as follows:

- A new mass-based approach for local outlier detection is introduced and experimented on both synthetic and benchmark datasets. This method utilizes the estimated mass function to measure the dissimilarity between instances instead of the distance-based measurement. Then, the outlier score function is constructed to compute the outlier's degree for each data point.
- The mass-based measurement has been exploited for the class imbalance problem. That alternative dissimilarity calculation is integrated into the weighted sum framework to predict the label of the query instance. As a result, a new mass-based similarity weighted k-neighbors for classifying imbalanced datasets is introduced and tested on 60 real-world application datasets. The experimental results show that our proposed model outperforms the other 11 competitive methods.
- Continuously exploiting the mass-based estimation with respect to the evidential reasoning theory, we introduced a new approach so-called EMass for solving the class imbalance problem. The experiments were conducted on a wide range of application domains, with different imbalance ratios, and a variety of instances. The experimental results show that our proposed methods outperform the other 12 competitive approaches. All results are validated by the nonparametric Wilcoxon signed-rank test.

1.2.3 Future directions

In future works, we continue to exploit the mass-based dissimilarity measurements among instances. These alternative functions can replace or merge with the distance-based or density-based functions in the conventional algorithms. It is not limited to the outlier detection and class imbalance tasks but also applying mass-based calculations for the other problems of machine learning, data mining, and knowledge discovery.

We acknowledge that this work is limited to compute on numerical data only. Therefore, we will extend the proposed models to categorical and mixing data in future works. In addition, we also plan to improve our models to detect the collective anomalies from multiple spatio-temporal datasets across different domains. Because when a collection of data points can be considered as outliers compared to the entire dataset and the individual instance might not be an outlier, is lacked.

Based on the successes of applying the Dempster-Shafer theory of evidence for the class imbalance task, we plan to formulate the mass function for the outlier detection problem by considering each neighbor of the query instance as an information source providing a piece of evidence to support

decision making.

1.3 Dissertation organization

This dissertation begins with the introduction to outlier detection, class imbalance challenges, the main research questions, significant contributions, and the directions for future works in Chapter 1. In Chapter 2 where the research background is prepared for understanding the whole dissertation. For example, the partitioning technique is briefly introduced first, then the mass-based dissimilarity measurement is reviewed with the main concepts and the definition on k lowest mass-based neighbors. The next two Chapters are the main body of this dissertation. Chapter 3 introduces a new alternative way for detecting local outliers. Chapter 4 presents two mass-based approaches for classifying the imbalanced datasets. Finally, Chapter 5 concludes this dissertation by summarizing the main points and directing several options for future studies.

Chapter 2

Research background

2.1 Hierarchical partitioning method

There are many partitioning methods to define the region for estimating the mass of an instance [27]. In this section, the isolation forest (IForest) is based on random tree algorithm to construct hierarchical partitions for a given dataset. The IForest technique was originally developed for anomaly detection as introduced in [28]. Figure 2.1 illustrates the hierarchical partitions result obtained by IForest.

The implementation of IForest includes two steps. The first step is to construct an IForest that contains a numbers of ITrees as a hierarchical partitioning structure. We use sub-sampling technique without replacement to create subsets from dataset $X = \{x_1, x_2, ..., x_n\}$. Then, we build each ITree for each subset respectively. To separate a sample set into two non-empty subsets, a random feature is selected at each internal node of an ITree, then split instances until every point is isolated or the maximum tree height is reached. The detail of the ITree building process can be found in [28] paper. The second step is for scoring the outlierness. However, in our work, the scoring step is replaced by evaluating mass-based score. This score represents the outlier degree. Then, we use this degree to predict the final outlier label. This step is presented in detail in the next section.

2.2 Mass-based dissimilarity measurement

On one hand, distance-based functions such as the Euclidean distance and Manhattan distance that have been commonly using to measure the dissimilarity between two instances in a given dataset. As a result, the distance calculations become the core operation in many tasks such as outliers detection and class imbalance problems in statistic analysis, machine learning, data mining, and knowledge discovery frameworks. However, those distance-based approaches might have shortcomings due to the strong assumptions on data-independent and distance axioms. In fact, the real-

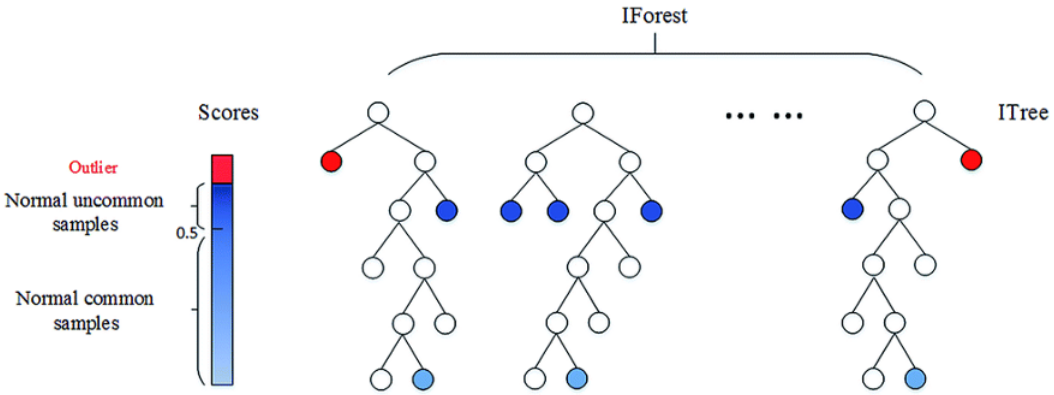


Figure 2.1: Isolation Forest (*Source* : *towardsdatascience.com*)

applications data are dependent naturally, and the distance axioms might not be satisfied if the datasets can not represent geometrically.

On the other hand, mass-based measurement offers a choice to provide the closest neighborhood match by employing estimation of the probability masses rather than the distances. Despite the widespread applications of the distance-based functions, research in psychology has pointed out since the 1970s that distance measures can not possess a key characteristic of dissimilarity as judged by humans, which two instances in a dense space are less similar to each other than two other instances with the same inter-point distance in a sparse space, according to Tversky 1977 [29], and Krumhansl 1978 [30]. These weaknesses of the distance-based algorithms or geometric models are redressed by using mass-based functions as introduced in [27,31]. The mass-based measures are used to replaced the distance-based measures directly, and the other components of the original algorithms are remaining the same. For example, Figure 2.2 shows the calculation results of the mass-based dissimilarity by referring to the Equation 2.2.

The following definitions present the main concepts for developing and deploying the mass-based dissimilarity measurement that was introduced in the previous study [32].

2.2.1 Definition 1

$S(x_t, x_i|H_j)$ or node S is defined as the smallest local node of the space that covers two given instances x_t and x_i with respect to the hierarchical partitioning structure H_j or isolation tree (ITree).

$$S(x_t, x_i|H_j) := \underset{S \in H_j, \{x_t, x_i\} \subseteq S}{\arg\max} \text{depth}(S|H_j) \tag{2.1}$$

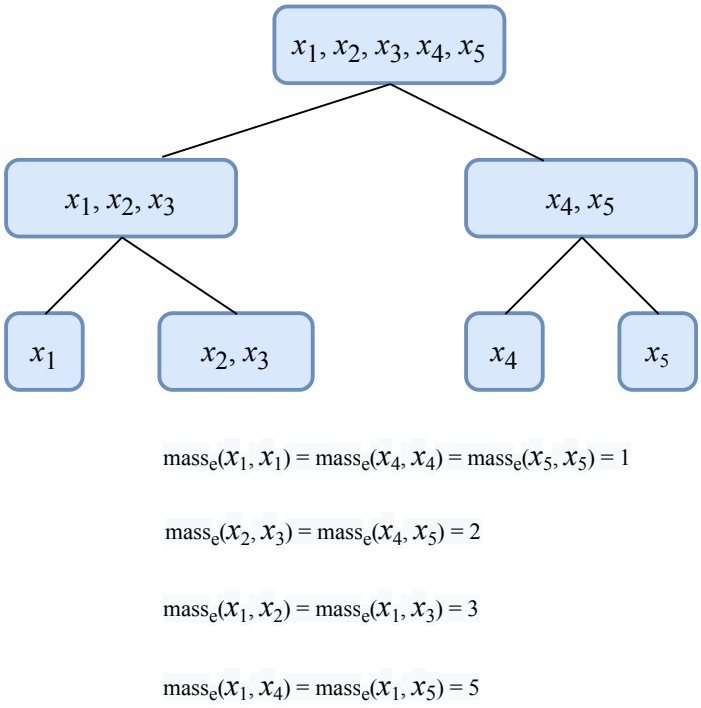


Figure 2.2: Results of calculated mass-based dissimilarity.

Where depth$(S|H_j)$ is the path length of node S in H_j or the number of partitions required to isolate node S from the root node in H_j.

2.2.2 Definition 2

The mass$(x_t, x_i|H_j)$ function is defined as the mass-based dissimilarity measurement between x_t and x_i, conditional on H_j. This is equivalent to the expectation of the probability that a random instance $z \in X$ belongs to the smallest space $S(x_t, x_i|H_j)$. This space is computed by equation (2.1) over all possible H_j, the so-called set $\mathcal{H}$.

$$\text{mass}(x_t, x_i|H_j) := E_{\mathcal{H}}[P(z|z \in S(x_t, x_i|H_j))]$$

In practice, we have h finite hierarchical partitioning structures H_j for a given dataset X. Thus, the mass$(x_t, x_i|H_j)$ is estimated as the average of the cardinality of $S(x_t, x_i|H_j)$ over the set $\mathcal{H}$.

$$\text{mass}_e(x_t, x_i) = \frac{1}{h} \sum_{j=1}^{h} \frac{|S(x_t, x_i|H_j)|}{|X|} \tag{2.2}$$

Notably that when the numbers of partitions, or ITrees, is infinity, the $\text{mass}_e(x_t, x_i)$ becomes the distance function between these two instances.

2.2.3 *k*-lowest mass-based dissimilarity neighbors

The context set of an instance x is defined equivalently to the set of the k-lowest mass-based dissimilarity neighbors around x, then called k-LMN(x) for short:

$$k\text{-LMN}(x_t) = \{x'_1, x'_2, ..., x'_k\}, \quad k \leq |X| \quad (\text{Cardinality of} \quad X) \tag{2.3}$$

$$\text{Where} \quad x'_i = \underset{x \in X \setminus \{x'_1, x'_2, ..., x'_{i-1}\}}{\arg\min} \text{mass}_e(x_t, x), \quad i = 1, ..., k$$

Note that the lowest probability mass neighbor (LMN) algorithms is another version of the nearest neighbor (NN) algorithms. This study shows that LMN algorithms overcome key shortcomings of NN algorithms in outlier detection and class imbalance tasks.

2.3 Dempster-Shafer theory

The Dempster-Shafer theory (DST) [33] of evidence is the most well-known framework for managing uncertainty and making decisions. In DST, Dempster's rule of combination to combine pieces of evidence is the central operation of the probable reasoning. Besides, the simple support functions and their corresponding discount are essential in the DST framework. In this study, DST is motivated to the class imbalance problem and the outlier detection task under perspectives of the rare and the uncertainty of the different instances belonging to the minority classes.

Based on the DST framework, we proposed an evidential classifier by considering each neighbor of a query instance as an information source providing a piece of evidence that supports predicting the class label. Then, each neighbor's posterior probability that belongs to its class is converted into a basic probability value (BPA) on the set of class labels. Dempster's rule of combination is utilized to pool these BPA values. The main idea of the evidential classifier is to try to contribute more important features to the objects that belong to the minority class.

We use a frame of discernment $\Omega = \{l_1, l_2, ..., l_M\}$ to denote a finite set of M class labels and the set of all subsets of Ω denoted as the power set 2^Ω. The basic probability (BPA) value is assigned for each member of 2^Ω. This BPA function is defined as, $m : 2^\Omega \rightarrow [0, 1]$ that satisfies the following conditions:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Omega} m(A) = 1 \tag{2.4}$$

Where function $m(A)$ represents a degree of belief on how strongly a subset $A \subseteq \Omega$ is supported by the piece of evidence. Each subset A such that $m(A) > 0$ is called a focal set of the function m. The degree of total ignorance is represented by $m(\Omega)$.

Dempster's rule of combination [34] is then applied to combine k BPAs m_i to obtain the evidence for determining the label of x_t. Mathematically, Dempster's rule is simply a rule for computing, from two or more BPA functions over the same set Ω, a new BPA function called their orthogonal sum ($\oplus$).

$$m_t(.) = \bigoplus_{1 \leq i \leq k} m_i(.)$$

Particularly, when combined by Dempster's rule, two BPA values $m_1(.)$ and $m_2(.)$ that are not conflicting and focusing on the same subset yield another BPA value, which can be computed as Equation 2.5.

$$m(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} \quad , A \neq \emptyset \tag{2.5}$$

Where $A, B, C \in 2^{\Omega}$ and $\sum_{B \cap C = \emptyset} m_1(B) m_2(C) < 1$.

2.4 Evaluation metrics

Evaluation metrics are a core component of building an effective machine learning model because of their capability to discriminate among model results. There are many different kinds of metrics to evaluate and compare the proposed models. Selecting the right metric depends on the type of data, model, and implementation plan. In this study, we attempt to introduce new approaches for outlier detection and class imbalance problem. It means we have been working with the imbalanced datasets. Therefore, we consider several following metrics that have been commonly using for learning from imbalanced datasets regarding to the literature.

First, a confusion matrix as the name suggests provides us an $N \times N$ matrix, where N denotes the number of classes. This matrix describes the complete performance of the model. For example, for the binary classification task, we have some instances belonging to two classes: YES or NO, $N = 2$ There is classifier which predicts a class for a given instance. On testing

this model on 165 instances, we achieve the following results as shown in Table 2.1.

Table 2.1: Confusion matrix

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50 (TN)	10 (FP)
Actual: YES	5 (FN)	100 (TP)

From the confusion matrix, we can compute accuracy score as represented in Equation 2.6, recall value as Equation 2.7, and precision value as Equation 2.8. For imbalanced datasets, the accuracy score could not asset well to choose a good model. Alternatively, the F1 score is a suitable measure instead. In this case, we try to obtain the best precision and recall at the same time. Hence, the F1 score presents the harmonic mean of precision and recall values for a classification task. The formula for the F1 score is as,

$$Accuracy = \frac{TruePositive(TP) + TrueNegative(TN)}{TotalInstances} \tag{2.6}$$

$$recall = \frac{TP}{TP + FN} \tag{2.7}$$

$$precision = \frac{TN}{TN + FP} \tag{2.8}$$

$$F1 = (\frac{recall^{-1} + precision^{-1}}{2})^{-1} \tag{2.9}$$

In addition, the area under the receiver operating characteristic curve (AUC-ROC) and the precision-recall curve (AUC-PR) are accepted metrics used for imbalanced datasets. The advantage of using the ROC curve is that it is independent of the change in the proportion of responses. The AUC-ROC presents a single number for the curve.

Finally, cross-validation is a significant concept in any data modeling technique. This concept tries to leave a sample on which we do not train the model but test the model on this sample before selecting the model. We apply k-fold validation with k=10 as usual by dividing the entire dataset into 10 equal parts. The proposed models are trained on nine parts and validated on the remaining part of the dataset. In 10 iterations, we have built a model

on each part and held each of them as a validation part. Then, we take an average of the results to find which of the models is best.

2.5 Non-parametric statistical analysis

When a distribution does not require to meet the significant assumptions to be analyzed, and the numbers of sample is small, a non-parametric test has been applied. Due to this reason, the non-parametric tests are referred to as distribution-free tests. Note that non-parametric tests provide an alternative way to parametric statistical analysis such as T-test or ANOVA. These parametric tests are valuable only if the underlying data satisfies the assumptions.

In this study, Wilcoxon signed ranks test [35] is used as a non-parametric statistical analysis to validate the experimental results of multiple pairwise comparisons. In the Wilcoxon test, the sum of the ranks for results of each comparing method versus the proposed approach. Because the experimental results satisfy the situations in which the application of non-parametric tests is appropriate. For example, the output data of the proposed models is often skewed distribution, or the size of the sample is too small. It may not be able to validate the data distribution. In addition, the output data may be ordinal or nominal.

Chapter 3

Outlier detection

As of January 2021, an extended version of this chapter is published at the IEEE Access Journal.

3.1 Introduction

Hawkins [36] defined an outlier as an observation that deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism. It means that an outlier is an instance or data point that is significantly different from the remaining data. In the data mining and statistical literature, outliers are also named abnormalities, anomalies, discordant, or deviants. Outliers exist in most application domains due to the unusual generating data processes. Therefore, an outlier often contains insight into abnormal characteristics of the system. Detecting such system properties may provide valuable information for a specific application.

For example, a computer network often collects different data types about the operating system, traffic, or user actions. This network may have unusual behaviors due to malicious activities. The outlier detection task here is referred to as intrusion detection system [37, 38], or IDS for short. This IDS monitors the network and searches for suspicious activity and know threats, then sending up alerts when it finds such items. The IDS has a function that remains critical in modern enterprise.

In the finance domain, credit-card fraud detection [39–41] play a significant role in protecting sensitive information for customers like credit card number and personal information. Authorized agents collect the transaction data to analyze and detect unauthorized users. Those activities have been considering as outlier detection tasks.

In surveillance systems, sensors have been employing to track the activities and locations in various environments. Abnormal changes in the considering patterns may present interesting events. Therefore, event detection [42, 43] is a critical application of outlier detection solution in this situation.

In many healthcare systems [44–46], a variety of devices has been employing to collect the data, for instance, magnetic resonance imaging (MRI) scans, positron emission tomography (PET) scans, electrocardiogram (ECG), or personal area network (PAN) devices. An unusual data point may represent the disease conditions.

Additionally, big datasets of space and temporal about weather patterns, climate changes, or land-cover have been collected through satellites and remote sensing systems [47–49]. Such kinds of datasets provide insights into environmental resources and human activities. Hence, the outlier detection tasks are advantageous to detect any changes.

However, outlier detection tasks are challenging in statistical analysis, machine learning, data mining, and KDD fields. There has been vast literature previously developed to solve the outlier challenges. Those approaches include distance- or geometric-, density-, clustering-, ensemble-, and learning-based methods. The main difference between these methods is the dissimilarity measurements among instances to assess the outlier degrees.

Commonly, outlier detection based on distance or density functions has been studying by numerous researchers such as the k-nearest neighbors outlier detector, the local outlier factor (LOF) and its variants, and the local outlier probabilities model (LOOP). The above-mentioned methods rely on how isolated the instance is from its neighboring ones. However, such methods have the weaknesses like depending on the data independence and distance axioms. For example, when the actual data is dependent, then the assumption about data-independent is broken. When the dataset does not represent geometry formations, the measure assumptions also could not be fulfilled. The k-NN outlier detectors might not determine the local outliers correctly because the design is based only on the distance measures of the instances.

Instead of distance-based estimations, few researchers have studied mass-based measurements for detecting the local outliers. As a matter of that fact, this chapter investigates the outlier scoring method based on mass dissimilarity judgments. Equally important, the authors in [50] proposed the gravitation-based framework that merges the assessment of mass and the distance function of the objects from perspectives of class imbalance solutions. In this work, the authors introduced the mass concept for each data particle is determined simply as the number of data points represented in a particular region.

Particularly, we present a novel mass-estimated method for detecting the local outliers, that contains three steps, as graphically depicted in Figure 3.2. The first stage requires preprocessing data for the outlier detection task from multiple data sources. The second stage is partitioning the prepared datasets,

where the first step of the iForest approach [28] is restated to partition the dataset. This step produces a set ITree or isolated tree-like structures. Then, in the third stage, the outlier degree is determined for each individual instance by the mass-based dissimilarity measurement.

This study has been driving by the effectiveness and efficiency of manipulating the mass-based dissimilarity measurement to solve the weaknesses of geometric-based or density-based calculations used in almost neighbor-based outlier detectors previously. With attention to the situation of a dataset has a various density, due to two pairs of instances have similar distance measurements, they may have different dissimilarities. Therefore, the distance-based functions could not operate well on that kind of dataset.

Then, we examine a novel unsupervised learning method for detecting the local outliers by utilizing the mass-based dissimilarity measurement. This concept is introduced firstly in [51], the authors defined the mass-based dissimilarity between two instances. In the mass-based local outlier score (MLOS) approach, we improved this concept to asset the dissimilarity between a data point and its context set, the so-called probability set mass, or pmass for short. The pmass is used to define the mass-based local outlier factor (MOLF), then to introduce the MLOS. This approach scores the local deviation of an instance that relevant to its the neighbors. The main idea seems comparable to the LOF model [52], however, the output of our approach is a probability value. It is a more explainable ability than the LOF approach.

In the MLOS method, the neighborhood is captured by the neighbors surrounding the k-lowest estimated masses, whose mass-based measurements are extended to evaluate the local probability mass or pmass. By comparing the analytical extent of an instance to its neighbors concerning in the pmass, the examined points can be recognized as outliers, and this idea is similar to the LOF approach.

The methodology taken in this research is a comparative method tested on real-world data science problems, and the measuring methods have been introducing in the literature reviews. The original contributions of this work are following:

- We proposed a new approach based on mass estimation to detect the local outliers by connecting the hierarchical partitioning procedure and the mass-based function. The central contribution is to propose a new outlier scoring method in the set of problems holding a varied density of data points.
- We aim to introduce an unsupervised learning method for modeling the local outliers under perspectives of statistical analysis and the machine

learning field.

- The tests are conducted on both synthetic and real-application datasets to evaluate the new model.

The remaining parts of this chapter are arranged as follows: In Section 3.2, the outlier modeling problem was formulated. In Section 3.3, a short review of the relevant works is presented. In Section 3.4, the mass-based local outlier factor, or MLOF for short, and the mass-based local outlier score called MLOS, are acquired. In Section 3.5, the experimental study and analysis are conducted, that compared the MLOS model with those of competitive studies. Finally, the conclusion is given in Section 3.6.

3.2 Problem formulation

The outlier modeling task has been formulating by following the definition of an outlier scoring function. Let $X \subset \mathbb{R}^d$ be an input dataset, in which the data dimension is denoted by d. Study on unsupervised machine learning model for regional outlier detection given dataset $X = \{x_1, x_2, \ldots, x_n\}$, we attempt to propose a unique outlier scoring model $f : X \to \mathbb{R}$ that yields the value of $f(x_i)$ to represent the outlier status of a given instance $x_i \in X$. The bigger the value of $f(x_i)$ is, the higher probability that x_i is an outlier. The outlier score contains available information for decision-making within a specific application domain.

3.3 Literature review

This section presently reconsiders the relevant methods that consist of the k-NN method for detecting anomalies, the LOF approach and its modifications, the LOOP model, and the IForest model.

3.3.1 Geometric outlier modeling

The k-NN outlier modeling algorithm [53, 54] computes an outlier degree for each instance by calculating its distance. This type of data-independent measurement calculates the distance to the k^{th}-NN or all k-NNs within the average value. The geometric-based measurement means how faraway an instance has deviated from the remaining ones. How to turn the hyper-parameter k has a notable impact on the outlier degree, as in k-NN classification methods. These geometric-based approaches could not applicable for modeling the local outlier with a variety of densities of instances. For

example, given an instance $p(x, y)$, the distance between p and its k^{th}-nearest neighbors $p_k(x_k, y_k)$ means the k-distance of p and denotes as $d^k(p)$:

$$d^k(p) = \sqrt{(y - y_k)^2 + (x - x_k)^2} \tag{3.1}$$

In this manner, the LOF approach [52] is the most famous local outlier modeling technique. The outlier score of a considering instance factors the density of this instance related to the densities of its neighborhood instances. The "reachability distance" concept, or reach-dist for short, is introduced by calculating k-distances referring to Equation 3.2, and vividly depicted as Figure 3.1.

$$\text{reach-dist}(x, y) = \max\{k\text{-distance}(y), \text{dist}(x, y)\} \tag{3.2}$$

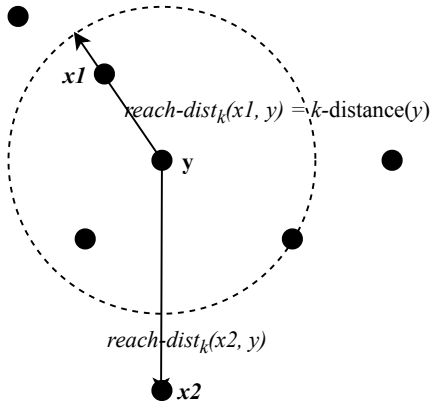


Figure 3.1: reach-dist(x_1, y) and reach-dist(x_2, y) for $k = 4$.

Note that y is a member of a set containing k neighborhood instances of x or k-NN(x). The k-distance(y) means the dist(y, o) distance between y and another instance $o \in k$-NN(y) satisfying (i) for at least k instances $o' \in k$-NN$(y) \setminus \{y\}$ it influences that dist$(y, o') \leq$ dist(y, o), and (ii) for at most $k-1$ instances $o' \in k$-NN$(y) \setminus \{y\}$ it results in dist$(y, o') \leq$ dist(y, o). The dist(x, y) distance means the distance between x and y. Therefore, reach-dist(x, y) is utilized to compute other measure, "local reachability density", so-called lrd(), referring to Equation 3.3:

$$\text{lrd}(x) = \frac{1}{k} \left(\sum_{y \in k\text{-NN}(x)} \text{reach-dist}(x, y) \right)^{-1} \tag{3.3}$$

Then, the measure of each instance lrd(x) has been compared to the lrd() states of its k-nearest neighbors. Again, the setting of hyper-parameter k is an essential step for this model, as in k-NN-based approaches.

Furthermore, the local outlier probabilities approach [46] linked the purpose of modeling a local outlier with a probabilistically oriented approach. This approach is based on distance relationships to assess the density of each instance. The LOOP approach solves the problems of the earlier models by employing probabilistic modeling and gaining a probability value referring to Equation 3.4, instead of the outlier score.

$$\text{LOOP}_S(x) := \max\{0, \text{erf}(\frac{\text{PLOF}_{\lambda,S}(x)}{\text{nPLOF}.\sqrt{2}})\} \tag{3.4}$$

Where the erf() notes the Gaussian error function for estimating the density of context set S of an instance x. The probabilistic local outlier factor is computed for instance $x \in X$ that concerns a significant λ and the context set $S(x)$ as introduced in [46]. This contextual neighbor identification approach has been driving this work. This context set is advanced in this dissertation to obtain the set of neighborhood instances for a considering one.

While comparing the outlier models on various datasets, the LOOP approach has more advantages than the others. For example, the ability to explain the results due to the output of the LOOP model is probability values. However, the LOOP method nonetheless depends on the distance computations.

Additionally, the LOOP method calculates the density deviation of an individual instance from its neighborhood instances. The main idea is closely akin to the LOF method. The LOOP method results in the probability values as outlier scores, where the one value represents the highest probability for the first outliers. The LOOP method determines a local density deviation based on the probabilistic set for each individual instance, with k pre-assigned nearest instances. In the same manner, as the LOF method, the LOOP method is a local outlier detection approach that depends on how each individual instance is scored concerning its neighbors.

3.3.2 Semi-supervised outlier modeling

In [28], the IForest approach had originally introduced for detecting abnormal instances using the hierarchical partitioning procedure. The key idea is that the unique instance is, the more possible it can be separated uniformly beyond some random attributes. The separation can have outliers isolate in one leaf node, then fewer separations can be chosen to detect the

outliers. This purpose can be formed to approximate pairwise dissimilarity by calculating the height of the ITree after which two instances are isolated.

In the isolation forest approach, the authors introduce the non-parameters outlier detector to operate efficiently with the high-dimensional datasets. For each subset of the given dataset, IForest built several trees so-called ITrees. Each tree is built by selecting a random attribute and random separating point to isolate instances until only one instance is obtained or the height of the ITree can be reached. Then, the outlier score is computed as follows:

$$f(x) = 2^{-E[\text{depth}(x)]/c(n_{\text{sample}})} \tag{3.5}$$

Where depth(x) is the length of the path that an instance x travels from the root node to the leaf one having x. $E[\text{depth}(x)]$ is the mean value of the depth(x) from a set of trees or an ITree. $c(n_{\text{sample}})$ is the mean depth of an unreachable search in a binary search. n_{sample} is the size of samples used to construct the ITree. When the $f(x)$ value is higher than 0.5, an instance x can be considered as an outlier.

From this work, the first step of the IForest method is integrated into the first stage of our proposed approach to partition a presented dataset. Then, the outlier scoring role in the IForest method is replaced by the new mass-based local outlier scoring function to achieve the outlier degree of each individual instance. In the detailed discussion about the outlier detection methods, we might mention surveying papers such as the following [41, 47, 55, 56].

3.4 Proposed MLOS approach

Various researchers have determined outlier scores in varied ways. Study on literature review, almost popular methods for scoring the outliers are based on distance or density calculation, as reviewed in the related works section. However, those methods have shortcomings in asset the local outliers due to a wide range of data points density.

This chapter is our main work, where we introduce a local outlier detection approach utilizing the mass-based estimation to compute the dissimilarity, then the so-called MLOS approach for short. Figure 3.2 shows the flowchart of the MLOS approach. This approach consists of three main stages. In the initial step, the input data have been preparing for detecting the local outliers. If the data labels are available, it can be separated and stored in the label spreading part. In the second step, the hierarchical partitioning technique such as IForest, which refers to Algorithm 3.1, is employed to isolate data points for the next step. In the third step, the

outlier degree or outlier scoring method is executed for each instance. This outlier scoring function is referring to the Algorithm 3.3.

Labeling outliers is an optional choice for evaluating the new MLOS approach, while this proposed approach is developed concerning the unsupervised learning method. Hence, the metric of accuracy is chosen to evaluate the proposed MLOS approach. To learn a new model for a specific application domain, the hyperparameters have been tuned by employing the cross-validation method.

An interesting mass-based measurement is the natural data-dependent dissimilarity measurement, and the distance axioms are relaxed. Therefore, the proposed MLOS approach could avoid the shortcomings of the distance-based or density-based outlier detectors.

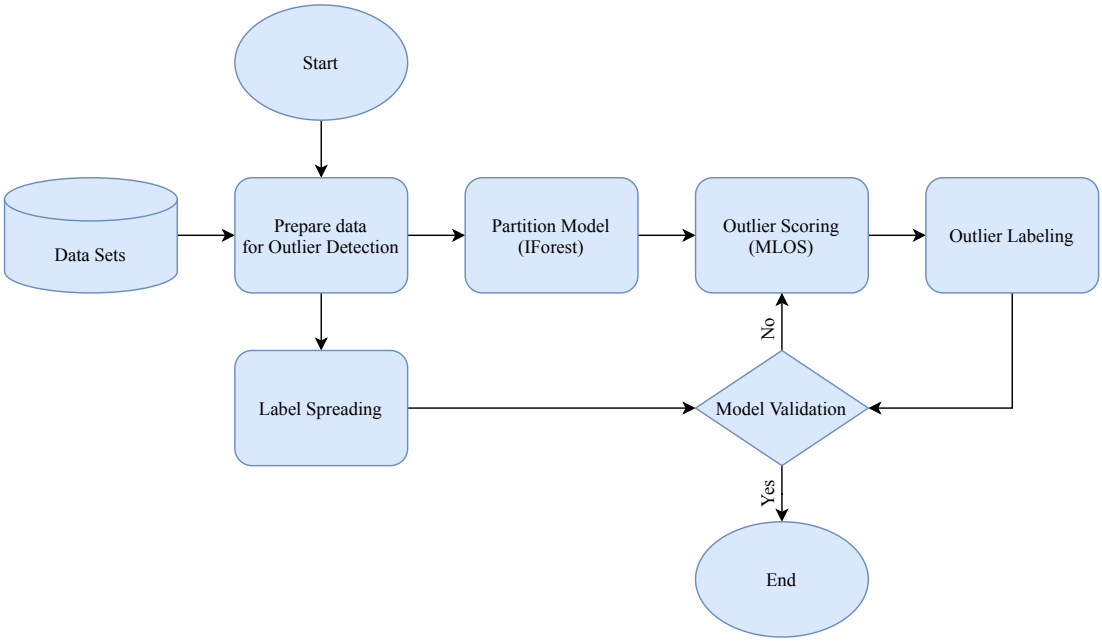


Figure 3.2: The proposed MLOS approach for detecting local outliers.

3.4.1 Notations

$X = \{x_1, ..., x_n\}$	$\triangleq$	Input data consists of n instances in $\mathbb{R}^d$
x, y, o	$\triangleq$	Given instances in X
t	$\triangleq$	Number of estimators or ITrees
X_j, $1 \leq j \leq t$	$\triangleq$	Subsets of X
$\mathcal{H} = \{H_1, ..., H_t\}$	$\triangleq$	Set of hierarchical partition structures of X
mass(.)	$\triangleq$	Function of the mass-based measurement
$S(x, y\|H_j)$	$\triangleq$	Smallest space in H_j covering x, y
$C(x)$	$\triangleq$	Context set of query instance x

Notably this study used the mass(.) function with the meaning of the density estimation, instead of the meaning of the mass function in the DST framework, where the BPA function is recalled.

3.4.2 Stage 1: Data preparation

The benchmark datasets are originally collected for multiple or binary classification tasks. Therefore, this stage does preprocess on the given datasets to prepare the "outlier" datasets. For example, the outliers are formed by the instances belonging to the minority class, and the inliers are the remaining instances. Before, we need to handle the missing data problem existing in a given dataset.

The data labels are separated and stored in the label spreading part. Then, these dependent variables will be used for validating the model after predicting the outlier labels.

3.4.3 Stage 2: Data partitioning technique

The effects of outliers on nearby data points may reduce by the subsampling technique, which uses to separate the original dataset X into multiple subsets or partitions. We observe that the outlier instances in a given dataset X tend to be outliers in the subsets where they present. In contrast, the inlier instances may be outliers in a few separation [28].

Algorithm 3.1 has executed on t subsets of input data X that are sampled with a size limit of n_{sample}. After that, the subset X_j is inferred to built an ITree or hierarchical partitioning composition H_j, $1 \leq j \leq t$. Practically, any hierarchical partitioning technique can be applied to build partition H_j. For the outlier detector modeling, the amount of estimator t and the sampling size n_{sample} should be large enough to treat all the instances in X; That

Algorithm 3.1 Data partitioning function

Input: input data X, estimators number t, size of sub-sampling n_{sample}
Output: $\mathcal{H}$: t hierarchical partitions
1: Partitions initialization $\mathcal{H} \leftarrow \emptyset$
2: Height limitation $l = \lceil \log_2(n_{\text{sample}}) \rceil$
3: **for** $j = 1$ to t **do**
4: $X_j \leftarrow \text{Random\_Sampling}(X, n_{\text{sample}})$
5: $H_j \leftarrow \text{Partitioning}(X_j, 0, l)$
6: **end for**
$\mathcal{H} \leftarrow \{H_1, \ldots, H_t\}$
7: **return** $\mathcal{H}$

means a data point x_i can be selected to be a part of at least one subset X_j. Hence, $tn_{\text{sample}} \geq n$ and for any $x \in X$, there is X_j satisfying $x \in X_j$.

3.4.4 Stage 3: Outlier scoring

After the partitioning data stage, a unique mass-based dissimilarity function for measuring the local outlier score so-called MLOS approach is proposed. This approach measures the outlier degree for each instance. In this approach, the first step of the isolation forest method is employed as the hierarchical partitioning technique due to the robustness and efficiency of the IForest framework. In the second step, finite hierarchical partition compositions $H_j, 1 \leq j \leq t$ are constructed from the given data X. Remark that the hierarchical partition H_j is a tree-like structured partition of the subset X_j, then so-called ITree:

$$H_j = \{S_1, S_2, ..., S_q\}, \quad q \leq |X_j|$$

Where S_i $(1 \leq i \leq q)$ represent the local nodes of each ITree. In this work, binary trees have been adopted for building the partitions.

It is important to realize how the MLOS approach is introduced, the following five concepts are defined as,

Definition 1. The smallest local node $S(x, y|H_j)$ are covering two instances x and y concerning to the ITree H_j as following:

$$S(x, y|H_j) := \underset{S_i \in H_j, \{x,y\} \subseteq S_i}{\arg\max} \text{depth}(S_i|H_j) \tag{3.6}$$

Where $\text{depth}(S_i|H_j)$ is the path length of node S_i in hierarchical partitioning H_j. The depth is defined as the number of separation steps that need to separate node S_i from the root of the ITree.

Definition 2. The mass-based dissimilarity function between two instances x and y is $\text{mass}(x, y|H_j)$ that conditions on H_j. The $\text{mass}(x, y|H_j)$ is equivalent to the probability expectation that an instance $z \in X$ randomly belongs to the space $S(x, y|H_j)$. That is computed by the equation (3.6) over $\mathcal{H}$.

$$\text{mass}(x, y|H_j) := E_{\mathcal{H}}[P(z|z \in S(x, y|H_j))]$$

Practically, there are limited t numbers of ITrees H_j constructed from the given data X. Therefore, the $\text{mass}(x, y|H_j)$ is estimated by the average of the cardinality of $S(x, y|H_j)$ on all possible H_j.

$$\text{mass}_e(x, y) = \frac{1}{t} \sum_{j=1}^{t} \frac{|S(x, y|H_j)|}{|X|} \tag{3.7}$$

Definition 3. The context set of instance x, $C(x)$, is defined as a set of k lowest mass neighbors surrounding x, the so-called k-LMN(x).

$$C(x) = \{y_1, y_2, ..., y_k\}, \quad k \leq |X| \tag{3.8}$$
$$\text{where} \quad y_i = \underset{y \in X \setminus \{y_1, y_2, ..., y_{i-1}\}}{\arg\min} \text{mass}_e(x, y), \quad i = 1, ..., k$$

Definition 4. Probability set mass of an instance x concerning to the context set $C(x) \subseteq X$ is defined with a significant λ as:

$$\text{pmass}(\lambda, x, C(x)) := \lambda\sigma(x, C(x)) \tag{3.9}$$
$$\text{Where} \quad \sigma(x, C(x)) = \sqrt{\frac{\sum_{c \in C(x)} \text{mass}_e^2(x, c)}{|C(x)|}}$$

is a standard of mass estimation.

The pmass values could be interpreted as the measurement of statistic extention of the context set $C(x)$ of instance x. To calculate the pmass value of each individual instance x concerning to its context set $C(x)$, the $\lambda = \sqrt{2}.\text{erf}^{-1}(\varphi)$ is utilized, instead of φ in the below property of the pmass, because the statistical extension induces some error:

$$\forall c \in C(x) : \text{P}[\text{mass}(x, c) \leq \text{pmass}(x, C(x))] \geq \varphi$$

Where erf represents the function of Gaussian error and $\text{mass}(x, c)$ equals the mass-based dissimilarity (Equation 2.2) between x and c, $\forall c \in C(x)$. This property indicates that the space surrounding x admits the pmass

boundary. This space also comprises any element of the context set $C(x)$ with a probability of φ. Algorithm 3.2 is a pseudo-code for the schemes used to calculate the pmass value for an instance.

Algorithm 3.2 Function of probability mass.

Input: $\lambda, k, x \in X, \mathcal{H}$
Output: pmass$(\lambda, x, C(x))$
1: **for** $y \in X, \quad y \neq x$ **do**
2: tmp $\leftarrow \emptyset$
3: **for** $H_j \in \mathcal{H}, \quad 1 \leq j \leq t$ **do**
4: calculate $S(x, y|H_j)$, //refer to Equation 3.6
5: $\text{mass}_e(x, y|H_j) \leftarrow |S(x, y|H_j)|/|X|$
6: tmp $\leftarrow$ tmp $\cup \{\text{mass}_e(x, y|H_j)\}$
7: **end for**
8: $\text{mass}_e(x, y) \leftarrow$ average(tmp), //refer to Equation 3.7
9: **end for**
10: $C(x) \leftarrow k$-MLN(x), //refer to Equation 3.8
11: **return** pmass$(\lambda, x, C(x))$, //refer to Equation 3.9

Eventually, the "3σ-rule" is a statistical rule to determine the outliers as the instance that differs more than λ times the standard deviation from the mean. The values of σ are achieved from the empirical studies, for example., $\lambda = 1 \leftrightarrow \varphi \approx 68\%, \quad \lambda = 2 \leftrightarrow \varphi \approx 95\%, \quad \lambda = 3 \leftrightarrow \varphi \approx 99.7\%$. The λ value depends on the nature of the dataset and specific application domain. Acknowing that the parameter λ may control properly the mass-based outlier scores. However, the λ implies just a normalization part that singularly affects the variation in the scores but does not affect the outliers ranking.

The authors in [46] presented the concept of the probabilistic distance, so-called pdist, that plays a significant component in developing the LOOP method. Motivated by this concept, we present in this dissertation the concept of probability set mass, the so-called pmass. Later, the pmass$(\lambda, x, C(x))$ computed for each individual instance $x \in X$ replaces the concept of pdist in the LOOP method. This simplistic replacement could address the weaknesses of the distance-based outliers detector. It also maintains the concept character of probability.

In definition 4, in the same manner with most statistic modeling approaches, we appropriate that the instance x plays the "center" of the context set $C(x)$, and the set of $C(x)$ generally obeys a Gaussian distribution. Then, $C(x)$ can be achieved by finding k lowest probability mass neighbors surrounding x, which refers to definition 3. The preferred assumptions are for calculating the standard value of masses.

Definition 5. The mass-based local outlier factor of an instance $x \in X$, the so-called MLOF, concerning to significant λ and the context set $C(x) \subseteq X$, is determined as follow:

$$\text{MLOF}(\lambda, x, C(x)) := \frac{\text{pmass}(\lambda, x, C(x))}{E_{y \in C(x)}[\text{pmass}(\lambda, y, C(y))]} - 1 \tag{3.10}$$

The MLOF value of an instance $x \in X$ measures the proportion of the pmass calculation surrounding x concerning the $C(x)$ and the expected values of the pmass of all instances in $C(x)$. This value is comparable with the LOF score as subtracting by one. As a result, a less than zero value means that the considering instance may not an outlier, while a greater value shows a rise in the likelihood that it may be an outlier. Comparable to the previous outlier detectors, for example, the LOF detector and its variants, these values could not be similar crossed various datasets.

The MLOS approach measures the outlier score of an instance as a probability value that an instance being an outlier or not. In addition, the MLOF measures are normalized by the aggregate value nMLOF, which is achieved during the computational method as,

$$n\text{MLOF} := \lambda\sqrt{E[(\text{MLOF})^2]} = \frac{\lambda\sum(\text{MLOF})^2}{n} \tag{3.11}$$

The nMLOF value might be interpreted as a standard deviation of the MLOF values. To turn the MLOF values into a probability value, we assumed that the MLOF values follow a normal distribution with the mean equal one with and the standard deviation is nMLOF. Therefore, the Gaussian error function has been computing to achieve the probability values. Hence, the final outlier score is determined by the following equation:

$$\text{MLOS}(\lambda, x, C(x)) := \max\left\{0, \text{erf}\left(\frac{\text{MLOF}(\lambda, x, C(x))}{n\text{MLOF}.\sqrt{2}}\right)\right\} \tag{3.12}$$

Finally, Algorithm 3.3 presents the proceeding for determining the MLOS score of an instance.

Algorithm 3.3 Function of Mass-based outlier score.

Input: $\lambda, x, C(x), \text{pmass}(\lambda, x, C(x))$
Output: $\text{MLOS}(\lambda, x, C(x))$

1: Calculate the pmass values for instances belonging to $C(x)$
$E_{y \in C(x)}[\text{pmass}(\lambda, y, C(y))] \leftarrow \frac{\sum_{y \in C(x)} \text{pmass}(\lambda, y, C(y))}{|C(x)|}$

2: Calculate the MLOF:
$\text{MLOF}(\lambda, x, C(x)) \leftarrow \frac{\text{pmass}(\lambda, x, C(x))}{E_{y \in C(x)}[\text{pmass}(\lambda, y, C(y))]} - 1$

3: Calculate the std.dev for MLOF values:
$n\text{MLOF} \leftarrow \lambda \frac{\sum (\text{MLOF})^2}{n}$

4: Calculate the MLOS:
$\text{MLOS}(\lambda, x, C(x)) \leftarrow \max\{0, \text{erf}(\frac{\text{MLOF}(\lambda, x, C(x))}{n\text{MLOF}.\sqrt{2}})\}$

5: **return** $\text{MLOS}(\lambda, x, C(x))$

3.5 Experimental result

The experimental results were achieved by testing the competitive models on both synthetic and real-world applications datasets. These performance results were compared among the proposed MLOS approach with the earlier competing outlier detectors. Later, the Wilcoxon signed ranks test was applied as a non-parametric statistical analysis to validate all the experimental results.

3.5.1 Experimental results on synthetic datasets

The MLOS approach was tested quickly on an artificially generated dataset, which was created by the Gaussian distribution sets centered at (-2,-2) and (2,2). The parameters of these two sets could be customized to ensure that there are various densities of instances in the datasets. This requirement is satisfied our considering scenario on the input dataset.

The outliers were attached by uniform distribution functions. In total, there are 47 instances with varied densities, as illustrated in Figure 3.3. To compute the MLOS values, the hyper-parameters such as down-sample size and number of ITrees are set as the same as in the IForest approach. Figure 3.3 to Figure 3.6 presents the results achieved by the proposed MLOS_, LOF_, IForest_, and LOOP_approach respectively.

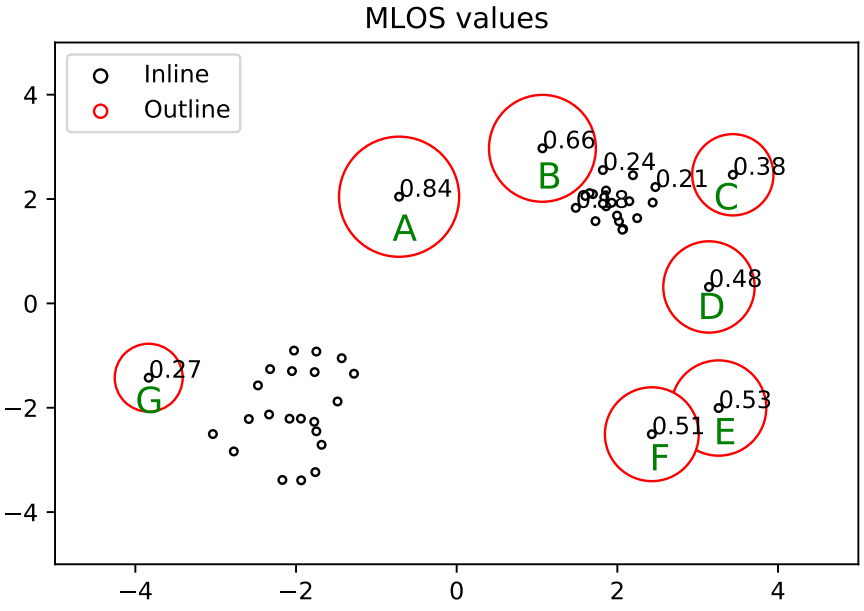


Figure 3.3: Results tested by MLOS approach.

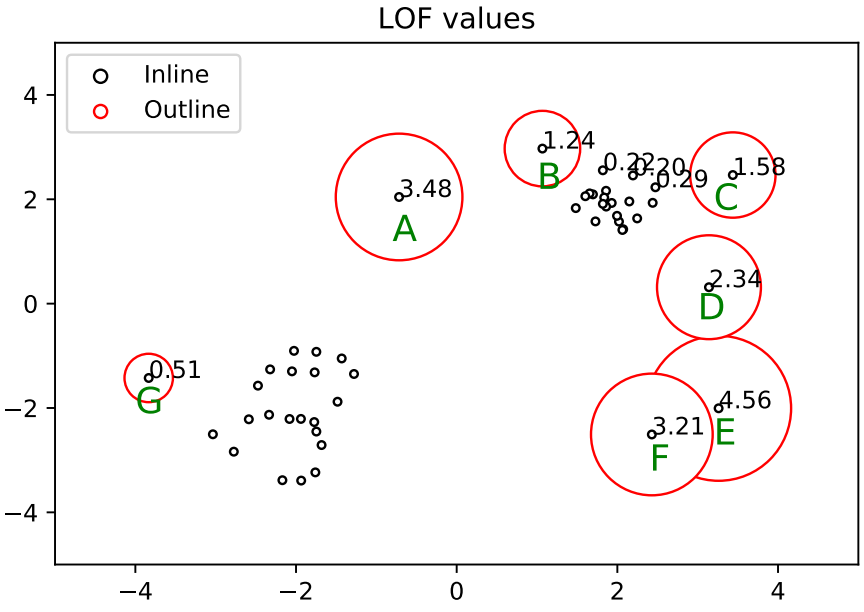


Figure 3.4: Results tested by LOF approach.

Results discussion: Seven green alphabet letters from A to G represent the detected outliers by our proposed approach, as shown in Figure 3.3. How big the value of the outlier score is representing by red circles that cover the detected instance and the inside number is the probability that instance is an outlier. A larger circle means a higher outlier score.

As we can see from Figure 3.3 to Figure 3.6 that all tested models detected the right outlier instances in the considering scenarios. Nevertheless, the ability to interpret these results is inconsistent among four competitive approaches. In the results of the LOF approach (Figure 3.4), instance A had a lower outlier score than instance E. Intuitively, the results may be interpreted inadequately.

In the results of the IForest approach (Figure 3.5), it is difficult to distinguish the scores of separate outliers. Because the identified outliers are presented by similar scores, slightly over 1/10. In the results of the proposed MLOS approach and the LOOP approach, the scores similarly confirmed that every single instance was an outlier regarding the probability outputs. Consequently, it was easy to interpret the obtained results. For example, instance A was significantly more than 80% likely to be an outlier based on either the MLOS or the LOOP approach.

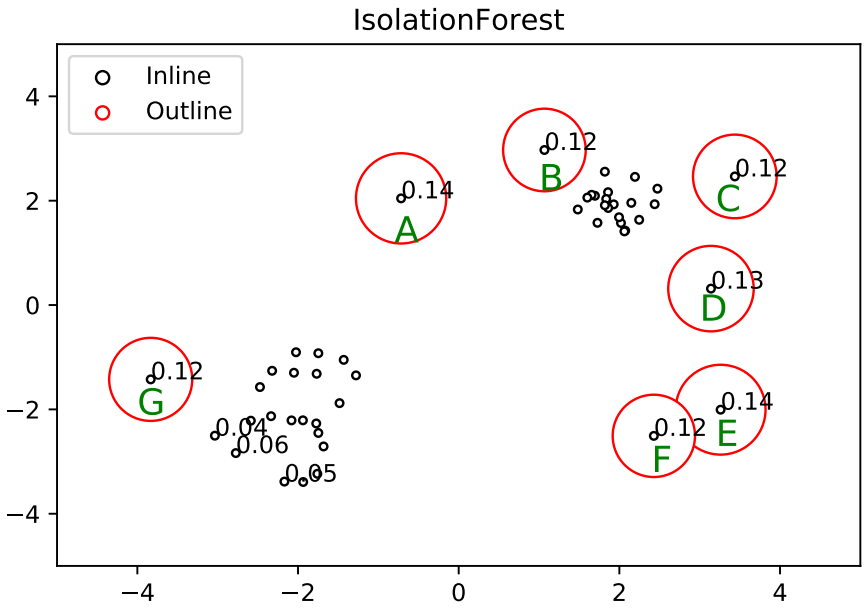


Figure 3.5: Results tested by IForest approach.

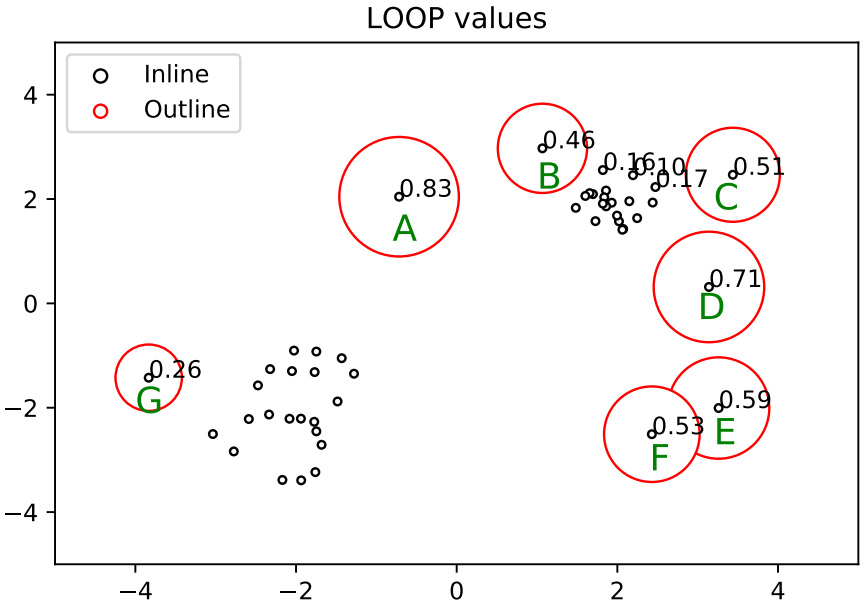


Figure 3.6: Results tested by LOOP approach.

As another illustration on synthetic datasets, the scikit-learn Python machine learning library [57] is employed to generate a new synthetic dataset that consists of five 2-dimensional datasets as illustrated in Figure 3.7. First, the *make-blobs* function creates the top-row datasets with various densities by allotting each class to one (as in the first row) or more (as in the second row and the third row) normally distribution sets of instances.

Second, the *make-moons* function generates two interleaving half-circle sets that include the optional Gaussian noise in the fourth row. The last artificial dataset is a random dataset generating by the *rand* function. Each type of these datasets included one or two modes to demonstrate the capabilities of the testing algorithms to manage multimodal datasets.

The outlier ratio or contamination parameter is set to 15%. It means 15% of the total instances are outliers. The experimental results achieved from these artificial datasets are displayed in Figure 3.7. This figure compares the performance of the MLOS approach with those of the baseline approaches, including the robust covariance or elliptic envelope algorithm, one-class SVM, IForest, and LOF algorithm.

The intuition behind the first three algorithms is straightforward. We draw the boundaries around the data points based on some criteria and classify any data point inside the borders as an inlier (black ones) and any

observation outside the boundaries as an outlier (red ones). On the other hand, the last two algorithms (LOF and MLOS) do not have borders because they have no prediction function.

Results discussion: All competitive models were tested on five synthetic datasets of 2-dimensions as displayed in Figure 3.7. The results are compared to each other. The determination borders between outliers and inliers are visualized in green color, except for the LOF and MLOS approaches.

For example, considering the instances in blue rectangles denoted by the uppercase letters A, B, C, and D. Firstly, instance A was recognized as a local outlier by the MLOS approach. However, the different approaches (the one-class SVM, IForest, and LOF ones) missed to capture it. Secondly, two instances belonging to rectangle B were calculated and identified variously, as outliers by the LOF approach only and as inliers by all other ones. Thirdly, the instances belonging to zones C and D were recognized as outliers only by the new MLSO approach, regarding the abilities of these models to detect the local outliers.

Until now, the MLOS approach is compatible with the other competitive approaches in this experiment. Additionally, the one-class SVM method [58] has been known to be sensitive to the outliers. This method did not perform well to detect the local outliers. This detector may be a benefit for detecting novelties when outliers do not exist in the training set. Modeling the local outliers play significantly challenging with high-dimensional datasets or without any underlying assumptions about the data distribution. In these scenarios, the one-class SVM method performs well based on how to turn the hyperparameters.

Equally important, the elliptic envelope or robust covariance algorithm operated under the assumption that the input datasets obey Gaussian distribution. Then, this algorithm learned an ellipse to separate the inliers and outliers. If the given data does not unimodal, this method could not perform well. Overall, this outlier detector is robust for the task.

Besides, the IForest and LOF approaches performed well for multimodal datasets. The power of the LOF approach over the other detectors, except the MLOS one, was recorded for the third data, where the data clusters contained various densities of instances.

Lastly, it is ambitious for decision making which approach is the best on the final artificial data. Notably that the one-class SVM method was lightly overfitted, and all detectors showed reasonable solutions for the aforementioned scenarios. There are intuitions on the proposed MLOS approach and

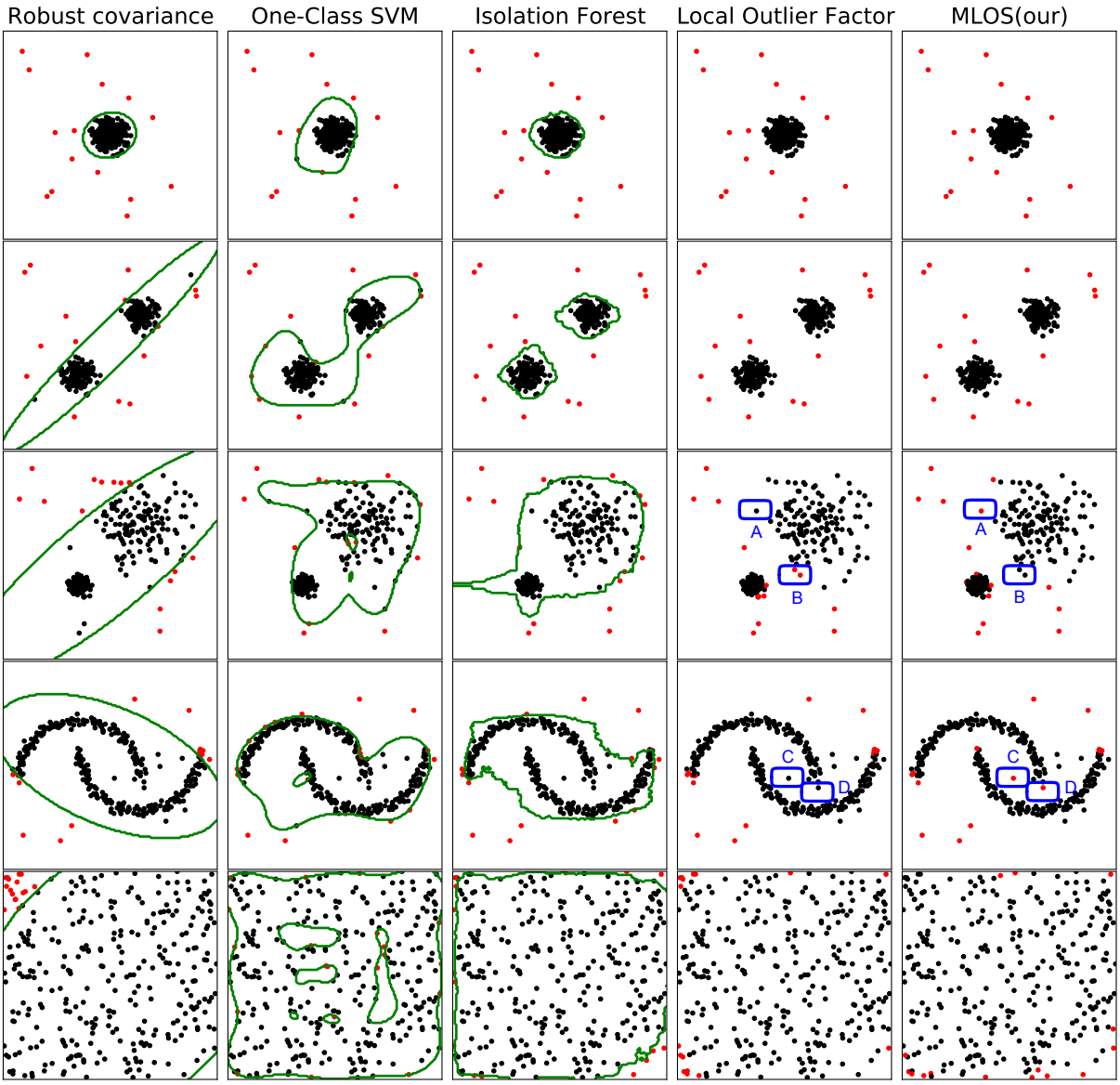


Figure 3.7: Outlier detectors comparison tested on five 2D datasets.

the other competitive ones when these experiments were conducted.

3.5.2 Experimental results on benchmark datasets

The experimental results were achieved by testing all competitive approaches on 25 benchmark or real-application data. These datasets have been experimented with in the earlier studies, to assess the performances of comparing outlier detectors. These benchmark datasets have been collecting from the knowledge extraction based on evolutionary learning (KEEL) [59], the UCI machine learning repositories [60], and the competition datasets from the Kaggle.

Those benchmark datasets were first collected for binary or multiple classification tasks. For modeling the local outliers under viewpoints of unsupervised machine learning, the given datasets need to prepare to fit into the models. This task can be obtained by considering the minority classes as outlier class and the remaining instances formed the inlier class. This technique was referred to as the outlier detection datasets (ODDS) organization. Table 3.1 presents the statistics and characteristics of the benchmark datasets in order by numbers of data points.

Table 3.1: The property of 25 real-application datasets for detecting outliers.

Datasets	Inliers	Outliers	#Points	#Dim.	#Outliers (%)
Appendicitis	0	1	106	7	21 (0.198)
Ecocardiogram	survived	died	106	9	32 (0.302)
Iris	Virginica, Versicolor	Setosa	150	4	50 (0.333)
Hepatitis	live	died	155	17	32 (0.206)
Wine	1, 3-10	2	178	13	71 (0.399)
Parkinsons	healthy	disease	195	22	48 (0.246)
Glass	1,2,3,4,5	6	214	9	9 (0.042)
Ecoli	others	omL, imL, imS	335	8	5 (0.015)
Bupa	2	1	345	6	145 (0.420)
Ionosphere	good	bad	351	35	126 (0.360)
Movement-libras	normal	irregular	360	90	6 (0.017)
WBC	bengn	malignant	378	33	21 (0.056)
Pima	0	1	768	8	268 (0.330)
Letter-Recognition	n	o	1,600	33	100 (0.063)
Cardio	normal	pathologic	1,831	22	176 (0.096)
Speech	American	others	3,686	400	61 (0.017)
Abalone	M, I	Other	4,177	8	1,307 (0.313)
Shuttle	1	2,3,5,6,7	4,646	10	90 (0.019)
Satellite	others	2,4,5	5,025	36	75 (0.015)
Banana	-1	1	5,300	2	2,376 (0.448)
Pen-local	1-9	0	6,724	17	10 (0.002)
Annthyroid	normal	hyperfunction	6,916	22	250 (0.036)
CreditCard	normal	fraud	28,432	30	49 (0.002)
KDDCup99	normal	attack	48,113	30	200 (0.04)
Aloi	n	o	50,000	28	1,508 (0.03)

We begin with one of the common life-threatening abdominal emergency datasets called "Appendicitis". That was introduced in the Computer Systems That Learn (1991) by S. M. Weiss, and C. A. Kulikowski. This dataset describes seven medical tests conducted on 106 patients. In which the class label stages if the patient has negative (class label 0) or positive appendicitis (class label 1). Class-1 formed the outlier as it is a minority class that includes 21 data points. The other class is considered as the inliers.

Another life-threatening crisis is heart strokes. People suffering heart strokes at some time in the past were gathered in the "Echocardiogram" dataset. In this data, the still-alive and survival people were grouped to indicate whether a patient survived for at least one year after the heart stroke. Those people were recognized as belonging to the normal class, while the remaining patients formed the outliers.

The "Iris" dataset is a set of multiple quantities in taxonomic issues for measuring the morphological varieties of the iris flowers including three relevant species. This dataset consists of 50 instances from each of three species: Setosa, Virginica, and Versicolor. The outliers were formed with one iris specified in each sub-dataset individually. For example, the Iris-Setosa was considered as the outliers, the remaining Iris-Virginica and Iris-Versicolor were considered as inliers.

The "Hepatitis" dataset contains occurrences of hepatitis in people. Hepatitis leads to infection of the largest inside a human organ, the liver, caused by the viral virus. The hepatitis viruses create dangerous diseases; for example, type B and type C, especially lead to chronic diseases. This dataset has 155 instances represented in 17 attributes with 32 people who were died from the hepatitis virus.

The "Wine" data is recorded about red and white alternatives of the Portuguese "Vinho Verde" wine. The excellent or poor wines were considered as the outliers due to limited excellent or poor wines than the normal ones.

The "Parkinson" dataset comprising biomedical voice analyses, which were gathered by 31 people, having 23 Parkinsons. The voice measures were recorded from those people to generate 195 corresponding instances represented by 22 attributes. Parkinson's state was considered as the outlier, and the healthy ones formed the normal or inlier class.

The "Glass" identification contains the dropped glasses that had nine attributes concerning different types of glasses. This dataset was utilized as evidence by the analysts. For the outlier detection task, class-6 was labeled as an outlier class since this class consists of only nine instances. In contradiction, the remaining 214 instances that belong to class-1 to class-5 have set inlier class.

The "Ecoli" dataset includes 7 numerical variants, and the "sequence

name" variant was removed from the "Ecoli" dataset. The outliers were composed of three classes: the omL, imL, and imS classes. The inliers were set by all the other majority classes.

The "Bupa" liver diseases dataset that consists of 345 instances with 6 features. The last column describes the presence or absence of liver disease, meant by value-2 and value-1, individually. Those values were utilized to form the outliers and inliers.

The "Ionosphere" dataset was formerly collected for binary classification tasks within 34-dimensions. This dataset can be downloaded from the UCI website. One variant having all zero values was eliminated. The outliers were set by the "bad" class and inliers were set by the "good" class. The proportion of outliers to the total instances is 126/351.

The WBC or "Wisconsin-Breast Cancer" [61, 62] has 357 "benign" and 212 "malignant" medicinal diagnoses that were reported with 33-dimensions by Dr. William H.W., Wisconsin, USA. On one side, the down-sampling technique was applied to the malignant class resulting in 21 outliers. On the other side, the inliers were formed by the benign class. There are 378 records within 5.6 percentage of outliers were prepared for this experiment.

The "Pima" set was first recorded by the National Institute of Diabetes, Digestive, and Kidney Diseases to forecast whether a patient is likely to ought diabetes. The binary classification task was considered a binary decision-making challenge. The patients were 21 years old females at the time experimenting. There are 768 instances in 8-dimensions. This Pima data includes 268 outliers (35% outliers).

A multiple classes dataset called the "Letter-Recognition" [63,64] includes the English alphabet that was represented by black-and-white rectangular visions with 16-dimensions. The inliers were formed by the sub-sampling technique from three letters, and then two sampled letters were merged randomly. There were 1,600 instances including 6.25% outliers, and these were processed in 32-dimensions for testing outliers detectors.

The analyses of fetal heart rates and uterine contraction characteristics from cardiography, so-called "Cardio", was tested for detecting the local outliers. There are three classes including suspect, normal, and pathology class. For the outliers detection task, the "normal" subset is considered as the inliers. The pathology class was downsampled to 176 instances to form the outliers. The instances belonging to the suspect class were eliminated.

Another real-dataset is the "Speech" set. This dataset consists of 3,638 segments of varying accents from the English language. The speech fragments are described by 400-dimensional attributes. The American accent formed the inliers, and only 1.65% corresponding to one other accent formed the outliers.

The "Abalone" dataset includes physical measures for estimating the age of abalone by following steps of cutting the shell through the cone, staining that, and then determining the number of rings with a microscope. Other alternative measures have been used to predict the ages by applying machine learning techniques. Hence, the abalone dataset was gathered for machine learning purposes. In this experiment, we considered each sex class as an outlier with slightly different portions.

The next dataset is called the "Shuttle" set represented by 9-dimensions features. For this "Shuttle" set, class-4 was eliminated, then class-2 to class-7 were merged to form the outlier class, while class-1 was considered as the inlier class. There are 4,646 data points were made with exactly 1.9% outliers in total.

The "Satellite" dataset was represented by 36-dimensional features. Three classes including class-2, class-4, and class-5 were merged to form the outlier class, which contains 75 data points. Besides, all the other instances belonging to class-1 and class-3 were combined to form an inlier class that consists of 4,950 instances in total.

The "Banana" dataset, which has two features only. The original binary classification task is to estimate an instance is whether it belongs to class -1 or class 1. In our experiment, class 1 was considered as the outliers and the other class as inliers. This Banana set has the highest outlier fraction in this work.

The next benchmark set is the "Pen-local", which includes 6,724 instances. There are 16 integer features and ten classes. For the outlier detection modeling, the number of instances in one class was decreased according to the digit "0", by a portion of 10%. Hence, 6,724 instances within ten outlier classes were made for this work.

The "Annthyroid" set was selected from the UCI website as the next dataset. There are three classes including hyperfunction, subnormal functioning, and normal. This data was represented by 15 categorical and six numeric attributes to decide whether a patient related to the hypothyroid. For this experiment, the subnormal class and the hyperfunction class together were treated as outliers and the normal class was formed the inliers.

Credit card fraud detection plays a significant role in finance and banking services. This real application dataset could be downloaded from the Kaggle website. For this experiment, the original dataset [39, 65–68] contained the transaction by European credit-card holders in September 2013. This dataset is extremely imbalanced since the positive class (fraud) accounts for approximately 0.02% of the total 28,432 transactions. It has only numerical input attributes. The 'Class' attribute reacted to the target, as it held values of one if fraud and zero otherwise.

The KDDCup99 dataset includes 34 continuous and 7 nominal features. In this experiment, 41 features were decreased to four ones including duration, service, duration, dst-bytes, and src-bytes. Using the nominal features 'service', this dataset was separated into SMTP, HTTP, FTP-data, and other ones. Then, we used the 'HTTP' service subset only, and all other subsets were eliminated. In total, there are 48,113 instances represented by 30-dimensions, including 200 outliers were prepared for detecting the local outliers.

Finally, the "Aloi" dataset was collected from the Harvard Dataverse database. That contains the highest number of instances used in this experiment. There are 50,000 instances represented by 28-dimensions features.

Results discussion: In this experiment, the MLOS approach was compared with other competitive outlier detectors that includes the IForest_ [28], LOF_ [52], LOOP_ [46], and density peak-based clustering approaches [69] so-called Den-Peak in this experiment. These approaches have experimented on a total of 25 real-application datasets. Because all of them were based on k nearest neighbors, each model performance was evaluated with different k values. Then, the average results of accuracy achieved by tenfold cross-validation testing are presented in the following table.

Table 3.2: The average accuracy results tested on 25 real application datasets.

Dataset	IForest	LOF	LOOP	Den-Peak	MLOS
Appendicitis	**0.655**	0.642	0.642	0.802	0.642
Echocardiogram	0.579	0.566	0.642	0.696	**0.698**
Iris	0.627	0.507	0.560	**0.667**	0.613
Hepatitis	0.536	**0.625**	0.482	0.580	0.620
Wine	0.539	0.438	0.438	**0.601**	0.573
Parkinsons	0.600	0.631	0.672	**0.754**	0.713
Glass	0.925	0.934	0.935	**0.958**	0.935
Ecoli	0.931	0.924	0.970	**0.973**	0.924
Bupa	0.507	0.507	0.530	**0.577**	0.501
Ionosphere	0.741	**0.829**	0.825	0.638	0.675
Movement-libras	0.889	0.867	0.867	0.933	**0.956**
WBC	0.936	0.937	0.910	0.933	**0.944**
Pima	**0.661**	0.609	0.531	0.651	0.552
Letter-Recognition	0.891	0.931	**0.939**	0.936	0.928
Cardio	0.895	0.841	0.837	**0.900**	0.839
Speech	0.967	0.968	**0.969**	0.852	**0.969**
Abalone	0.564	0.541	**0.580**	0.451	0.575
Shuttle	**0.993**	0.966	0.966	0.980	0.962
Satellite	**0.987**	0.978	0.978	0.931	0.972
Banana	0.557	**0.798**	0.503	0.287	0.517
Pen-local	**0.997**	**0.997**	0.996	0.995	**0.997**
Annthyroid	0.944	0.940	0.938	0.036	**0.956**
CreditCard	0.997	0.996	0.996	0.996	**0.998**
KDDCup99	0.988	0.980	0.984	0.010	**0.998**
Aloi	0.944	0.948	**0.951**	0.933	0.934
Avg. Values	0.794	0.796	0.786	0.723	**0.800**
Avg. Ranks	2.760	3.000	3.040	2.920	**2.600**

Overall, the new MLOS model obtained the largest value of the average accuracy and the best average rank. It means that the MLOS outperformed the other competitive approaches in the accuracy metric. Particularly, the MLOS approach outperforms the other competitors on seven sets including the Annathyroid, CreditCard, Echocardiogram, Speech, KDDCup99, Movement-libras, and WBC sets. Nevertheless, the metric of classification accuracy alone may typically not enough information to decide which approach is the best, especially for class imbalance like the local outliers detection task.

Table 3.3: The comparison of F1 score tested on real application datasets.

Dataset	IForest	LOF	LOOP	Den-Peak	MLOS
Appendicitis	0.847	0.776	0.776	**0.890**	0.776
Echocardiogram	0.770	0.689	0.743	0.722	**0.784**
Iris	0.560	0.630	0.670	**0.800**	0.610
Hepatitis	0.677	0.600	0.554	**0.734**	0.569
Wine	0.617	0.533	0.533	**0.751**	0.645
Parkinsons	0.735	0.755	0.782	0.754	**0.810**
Glass	0.961	0.961	0.966	**0.979**	0.956
Ecoli	0.991	**0.994**	0.985	0.986	0.982
Bupa	0.575	0.575	**0.595**	0.577	0.570
Ionosphere	0.831	0.831	**0.884**	0.779	0.787
Movement-libras	0.940	0.929	0.929	0.933	**0.976**
WBC	0.966	**0.975**	0.952	0.971	0.955
Pima	0.740	0.742	0.640	**0.788**	0.656
Letter-Recognition	0.942	0.953	**0.967**	0.965	0.935
Cardio	0.942	0.938	0.910	**0.947**	0.901
Speech	0.983	0.983	0.984	0.919	**0.985**
Abalone	0.683	0.662	**0.694**	0.451	**0.694**
Shuttle	**0.998**	0.991	0.983	0.990	0.980
Satellite	**0.994**	0.993	0.989	0.964	0.985
Banana	0.599	**0.800**	0.550	0.287	0.563
Pen-local	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**
Annthyroid	**0.971**	0.967	0.968	0.967	0.963
CreditCard	0.998	0.998	0.998	**0.999**	0.998
KDDCup99	**0.995**	0.990	0.992	0.989	0.990
Aloi	0.970	0.973	**0.975**	0.969	0.969
Avg. Values	**0.851**	0.850	0.841	0.844	0.842
Avg. Ranks	**2.640**	2.720	2.800	2.800	3.400

Therefore, the other unambiguous and clean evaluation metrics like using confusion matrix should be used to evaluate the results of the outlier modeling. Since the local outlier detection modeling could be viewed as a binary classification task. Then, the confusion matrix can be utilized to describes the breakdown of error types for each approach. On the other hand, the F1 score has been used to cover the balance between the precision and recall values that were calculated by the confusion matrix. Table 3.3 summarizes the average F1 values from the tenfold cross-validation tested on the 25 real datasets and compared them among the five approaches. In summary, the IForest approach obtained the best results in both the average values and average ranks. The MLOS approach achieved the best performances on particular datasets, such as the Echocardiogram, Movement-libras, Speech, and Parkinson's set.

In the meantime, for a given instance, obtaining the outlier score repre-

sented by a probability value is more flexible than predicting the outlier label directly. This probability value is considered as the "outlierness" degree of the query instance. In other words, the flexibility to interpret the results can be achieved by outputting the probability values.

In addition, the model operator can accept varying thresholds to maintain the trade-offs among the errors created by different testing models. For example, the false positives (FP) numbers are compared with the false negatives (FN) numbers. A threshold could be chosen to measure the execution performance of the model for a particular dataset. Then, two other diagnostic plots are used, including receiver operating characteristic (ROC) curves and precision-recall (PR) curves, to assist the interpretation. The ROC curves depict the trade-off between the rate of true positive and false positive. The PR curves describe the trade-off between the recall and precision of a model by using various thresholds.

The results from the area under the curve (AUC) for real datasets are illustrated in Table 3.4. Overall, the LOOP model performed the highest average values for both the ROC-AUC and PR-AUC scores, and the best ranking for the ROC-AUC. Though, the IForest approach obtained the best average ranks for the PR-AUC scores. Our proposed MLOS approach outperformed the other models on particular datasets, for example, the Speech and Hepatitis set, on both evaluation metrics.

Table 3.4: The comparison of AUC results tested on real application datasets.

Dataset	ROC-AUC				PR-AUC			
	IForest	LOF	LOOP	MLOS	IForest	LOF	LOOP	MLOS
Appendicitis	0.314	**0.479**	0.418	0.435	**0.187**	0.173	0.169	0.142
Echocardiogram	**0.600**	**0.600**	0.588	0.524	**0.401**	0.396	0.396	0.329
Iris	0.378	0.491	0.561	**0.510**	0.297	0.339	**0.444**	0.367
Hepatitis	0.559	0.424	0.462	**0.572**	0.423	0.370	0.381	**0.512**
Wine	**0.595**	0.518	0.426	0.504	**0.456**	0.400	0.339	0.417
Parkinsons	**0.723**	0.513	0.527	0.560	**0.383**	0.283	0.287	0.317
Glass	0.585	0.555	**0.725**	0.638	0.070	0.066	0.214	**0.245**
Ecoli	0.555	0.561	**0.789**	0.606	0.130	0.188	**0.271**	0.253
Bupa	**0.533**	0.483	0.513	0.516	0.425	0.399	0.417	**0.436**
Ionosphere	0.736	0.792	**0.901**	0.625	0.640	0.689	**0.858**	0.545
Movement-libras	0.294	0.367	**0.677**	0.369	0.044	0.051	**0.091**	0.033
WBC	0.668	0.467	**0.679**	0.637	**0.157**	0.056	0.126	0.147
Pima	0.473	0.505	0.481	**0.513**	0.378	0.337	0.334	**0.385**
Letter-Recognition	0.546	0.771	**0.907**	0.540	0.084	0.284	**0.581**	0.095
Cardio	**0.744**	0.491	0.539	0.559	**0.284**	0.107	0.126	0.233
Speech	0.469	**0.777**	0.607	0.503	0.015	**0.102**	0.044	0.017
Abalone	**0.529**	0.508	0.512	0.506	**0.332**	0.318	0.318	0.335
Shuttle	**0.952**	0.515	0.630	0.712	**0.168**	0.038	0.073	0.129
Satellite	**0.863**	0.525	0.759	0.656	0.208	0.064	**0.210**	0.067
Banana	**0.566**	0.494	0.496	0.503	**0.482**	0.453	0.444	0.456
Pen-local	0.689	0.429	**0.955**	0.559	0.003	0.005	**0.028**	0.003
Annthyroid	0.581	0.624	**0.755**	0.535	0.092	0.058	0.106	**0.120**
KDDCup99	**0.827**	0.548	0.711	0.594	0.060	0.023	**0.048**	**0.048**
Avg. Values	0.599	0.541	**0.635**	0.551	0.249	0.226	**0.275**	0.245
Avg. Ranks	2.261	3.043	**2.087**	2.609	**2.174**	3.174	2.348	2.304

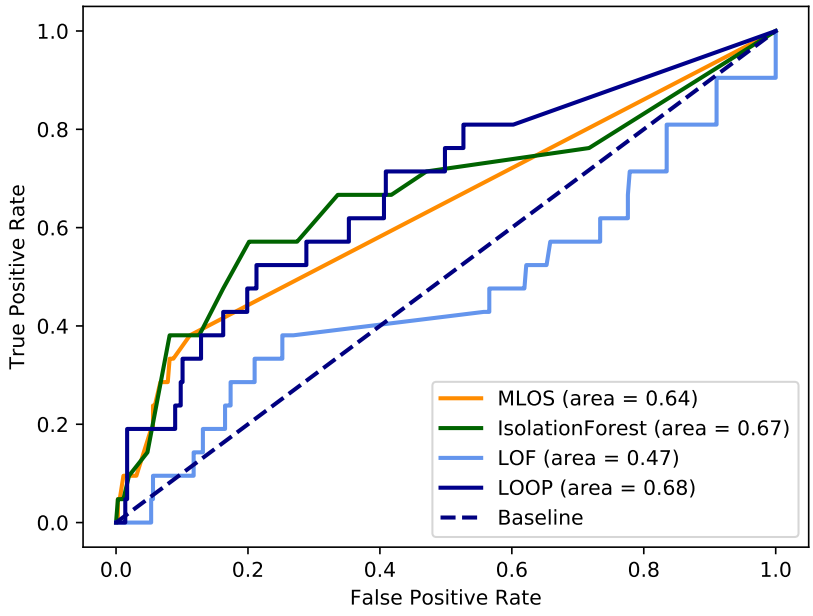


Figure 3.8: ROC curves tested on the WBC dataset.

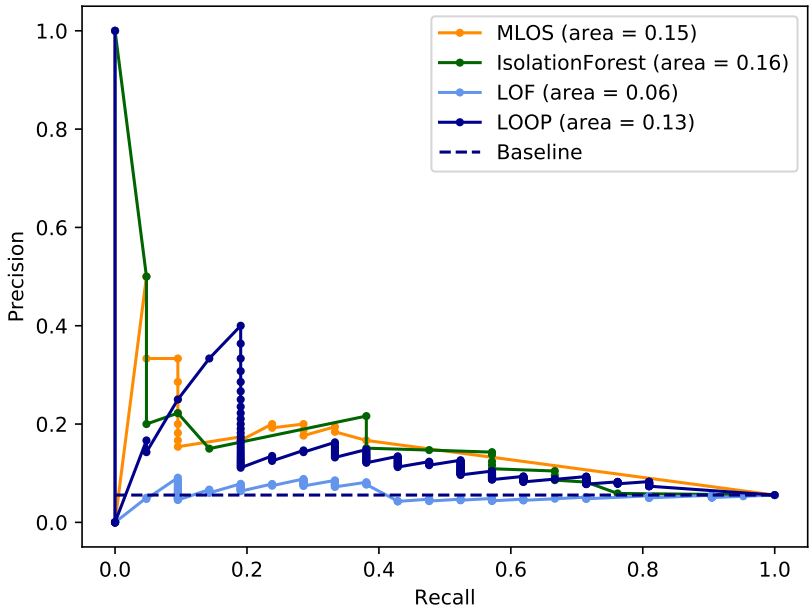


Figure 3.9: PR curves tested on the WBC dataset.

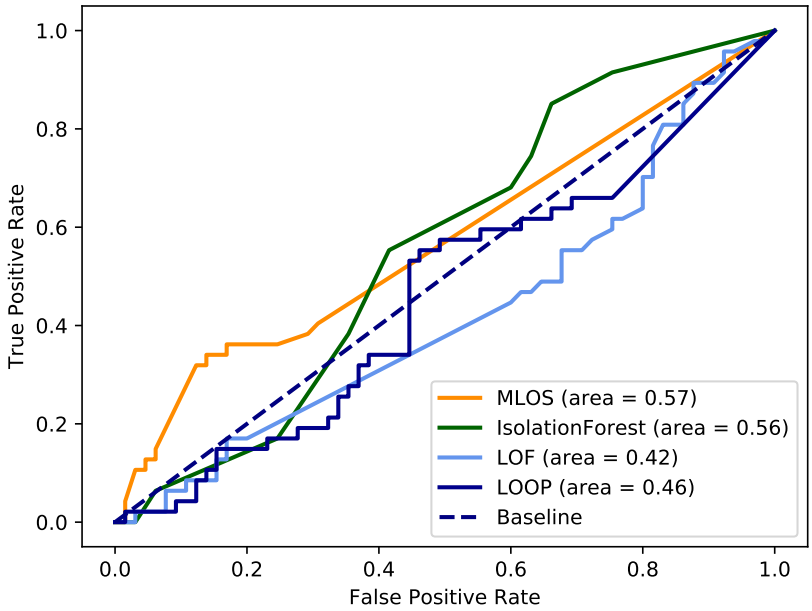


Figure 3.10: ROC curves tested on the Hepatitis dataset.

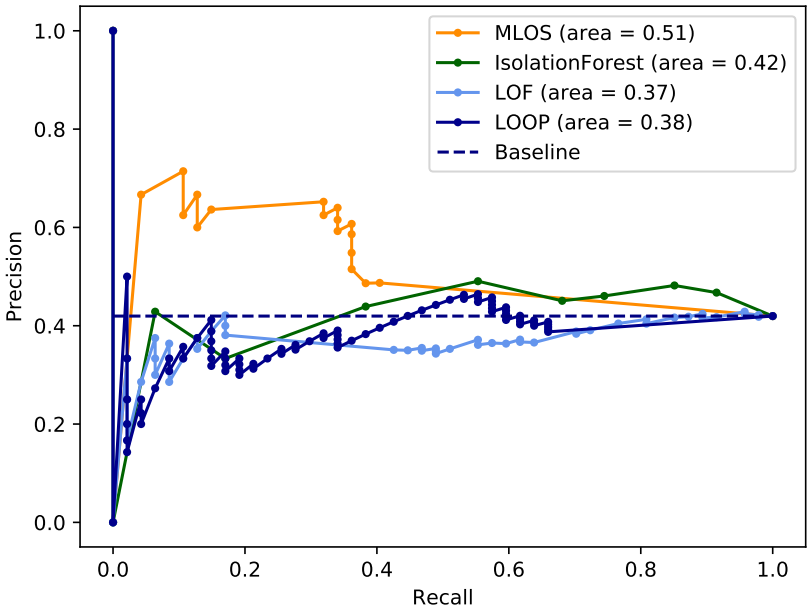


Figure 3.11: PR curves tested on the Hepatitis dataset.

The visualization of the ROC-AUC and PR-AUC results for particular datasets as illustrated in Figure 3.8, Figure 3.9, and in the appendix section. These two figures illustrate that the MLOS approach obtained the second-best on the ROC-AUC and PR-AUC scores for the WBC set. Figure 3.10 and Figure 3.11 illustrate that the MLOS approach achieved the best result for the Hepatitis dataset in both the ROC-AUC and PR-AUC scores.

3.5.3 Non-parametric statistic test

The experimental results need to validate, then the Wilcoxon signed ranks test [35] was chosen as a non-parametric statistical test to compare multiple pairwise approaches. Because the experimental results do not follow any distribution assumptions, and the experiments were conducted on a small number of datasets. Hence, the non-parametric statistical analysis was employed by using the IBM SPSS software that was supported by the research center for advanced infrastructure in JAIST.

The Wilcoxon signed ranks test outlines the sum of the ranks from the results of each comparing method versus the MLOS approach. The R^+ was denoted as the sum of the positive ranks and R^- as the sum of the negative ranks. The Wilcoxon signed ranks were tested on all the results of the experiment conducted on the real application datasets.

The Wilcoxon test results for multiple pairwise comparisons on the accuracy metric achieved by the MLOS approach and the other competitive ones are reported in Table 3.5. As we can see, the Den-Peak approach is the best one among the others due to the highest R^- values. However, the MLOS approach outperforms the Den-Peak according to the results of multiple approaches comparison reported in Table 3.2, and there are significant difference between the two approaches.

Table 3.5: Wilcoxon signed ranks test for accuracy metric.

MLOS vs	$\mathbf{R^+}$	$\mathbf{R^-}$	***p*-value**
LOF	687.5	347.5	0.055
LOOP	718.0	317.0	0.024
IForest	647.0	434.0	0.245
Den-Peak	652.0	623.0	0.889

The Wilcoxon signed ranks test results for comparing multiple pairwise approaches on the F1 metrics are revealed in Table 3.6. Though the IForest approach is the model obtained the highest ranking in the multiple comparisons on the real application datasets as reported in Table 3.3. The Den-Peak approach is the best one from the pairwise comparison due to it achieved the best R^- value here.

Table 3.6: Wilcoxon signed ranks test for F1 metric.

MLOS vs	$\mathbf{R^+}$	$\mathbf{R^-}$	***p*-value**
LOF	507	439	0.681
LOOP	526	509	0.924
IForest	485	596	0.554
Den-Peak	437.5	738.5	0.123

The Wilcoxon signed ranks test results for the ROC-AUC metric are reported in Table 3.7. As we can see, the LOOP approach obtained the best result among the others. This result is the same as the results of multiple comparison reported in Table 3.4. Additionally, the test shows a p-value of 0.46. On the other hand, the LOF approach is the worst among the others and the test show a p-value of 0.026, which is lower than 0.05. Hence, the MLOS approach outperforms the competitive one with a confidence level higher than 95% for the pairwise comparison.

Table 3.7: Wilcoxon signed ranks test for ROC-AUC metric.

MLOS vs	$\mathbf{R}^+$	$\mathbf{R}^-$	***p*-value**
LOF	804.5	371.5	0.026
LOOP	516.0	660.0	0.460
IForest	731.0	445.0	0.142

The Wilcoxon signed ranks test results for the PR-AUC metric are reported in Table 3.8. As we can see that the IForest approach is the best one comparing to the remaining approaches. In addition, the LOF approach is the worst one among the others, and the test show a p-value of 0.002, which is lower than 0.01. Hence, the MLOS approach outperforms the competitive ones with a confidence level higher than 99% for the pairwise comparisons.

Table 3.8: Wilcoxon signed ranks test for PR-AUC metric.

MLOS vs	$\mathbf{R}^+$	$\mathbf{R}^-$	***p*-value**
LOF	853.5	274.5	0.002
LOOP	689.5	438.5	0.184
IForest	556	525	0.866

3.6 Chapter conclusions

This chapter presents an alternative approach for detecting the local outliers concerning the scenario of varying densities of instances. This proposed model is based on a mass estimation to compute the dissimilarity among instances. This mass-based dissimilarity measurement combined with probabilistic modeling is exploited, instead of distance-based or density-based measures to compute the dissimilarity as in conventional outlier detectors. In the MLOS approach, the hierarchical partition technique is used to separate the input data. Then, the mass-based dissimilarity between each considering instance and its context set is computed. Thereafter, the proposed mass-based local outlier scoring (MLOS) function is employed to calculate the outlier score for each instance.

Optionally, the MLOS approach may predict the outlier label for each instance. This work also advanced the other important approaches including the LOF, IForest, and LOOP one for detecting the outliers in a given dataset.

There has been a vast of literature that deeply concentrated on labeling and scoring the outliers. Nevertheless, we are interested in the data nature and the interpretability of the outlier scores. Therefore, this work centers

more on the outlier scoring approach due to the ability to explain the results, and to convert scores to labels.

Additionally, to outline various modeling techniques for the local outliers detection, the two foremost dissimilarity measurement frameworks that involve data-independent metrics and data-dependent metrics are considered. These metrics are meaningful for "scoring" outlier modeling method. That results in probabilistic values. This value can be interpreted directly for detecting the outliers in varying real application domains.

In conclusion, although the outlier modeling problem has been studying broadly, the area itself is continuing by adding new datasets, issues, novel solutions or theories, software implementations, and many different insights. This work aims to introduce an alternative outlier approach that yielded comparable results. All the experiments were conducted on artificial and benchmark datasets. It also emphasizes that the mass-based measures can fix the shortcomings of distance-based dissimilarity calculations.

In future works, we try to combine the distance-based methods with the new mass-based approach. This combination will merge the geometric information into the mass-based estimation. We aim further improve the quality of the local outlier modeling technique. Next, we aim to exploit the measurement methods for rare and uncertain events by studying Dempster-Shafer theory for detecting the outliers. In this strategy, each neighbor of a given instance can be considered a piece of evidence commits to score the outliers. Then, Dempster's rule of combination or any other combination rules will apply to model the outliers. Finally, we will apply the new models to more real application domains.

Chapter 4

Class imbalance

4.1 Introduction

Generally, the classification, clustering and outlier detection are three main challenges for statistical analysis, machine learning, data mining, and knowledge discovery frameworks. This chapter focuses on the classification problem for imbalanced datasets. Given dataset has predefined categorical classes, classification involves determining which of these classes an unseen instance belonging.

There has been vast literature on a classification problem. In other words, a varied classification approaches have been proposed and successfully applied in a variety range of application domains, for examples C4.5 Decision Tree (DT) [70, 71], Naïve Bayes (NB) [72, 73], k-Nearest Neighbor (k-NN) [74, 75], Logistic Regression (LR) [76, 77], Random Forest (RF) [78, 79], Linear Support Vector Machine (LinearSVM) [80, 81], Gaussian RBF kernel SVM [82], Bagging algorithm [83, 84], Decision Trees with AdaBoost [85, 86], XGBoost [87, 88], and Gaussian mixture model Proximity Weighted Evidential (mPEkNN) method [89].

However, most of these approaches focus on balanced datasets. They do not directly classify the imbalanced datasets. In an imbalanced dataset, the minority class has only a small portion of all the instances, while the majority class has a large one. Hence, the classification tasks become more challenging problem on the imbalanced datasets because of the skewed class distribution while classifiers treat all instances equally. In addition, a varied density of data points poses another significant challenge that distance-based or density-based classifiers cannot perform well.

These two main challenges require either change the dataset or specialize the learning algorithm to handle the class imbalance task. The preceding studies commonly focus on four groups of methods to handle the class imbalance challenges. These methods include algorithmic modifications, resampling data space, cost-sensitive classification, and ensemble learning. Firstly, the algorithm-oriented approaches develop new algorithms or adapt

existing ones for the class imbalance problem. Secondly, the resampling techniques, which preprocess the data to reduce the effectiveness of their class imbalance, include the over-sampling method like the synthetic minority oversampling technique or SMOTE for short [90], and under-sampling methods like the Tomek links algorithm [91]. Thirdly, the cost-sensitive learning solutions incorporate both the algorithmic and data-level approaches to decrease misclassification costs for the minority class. Lastly, the ensemble learning methods are conducted either by embedding a cost-sensitive framework in the ensemble learning process or modifying the existing ensemble algorithms at the data level approaches.

The limitation of these learning methods is that the instance from each class are treated equally, it means that the learning algorithms consume more resources to update learnable parameters for one class than another for imbalanced datasets. Another limited point of those classifiers involves the varying density of instances in the given datasets.

In this chapter, we present two new approaches for the class imbalance tasks. These approaches can overcome the imperfections of the distance-based or density-based classifiers. The first proposed classifier is a new mass-based similarity weighted confidence k-neighbor approach, so-called Sk-LMN for short. The experimental results show that Sk-LMN outperforms the other 11 competitive models tested on 60 imbalanced datasets in terms of the precision-recall (PR-AUC) metric. The F1 score is used for multiple comparisons 12 tested models as well. In addition, the Wilcoxon signed ranks tests are employed as non-parametric statistic analysis to validate all the experimental results.

The second proposed classifier is a new mass-based similarity integrating with evidential reasoning approach, so-called EMass, for the imbalanced datasets. The advantages of mass-based similarity measurement and Dempster-Shafer theory of evidence are strong motivations for us to exploit them for the problem of imbalanced classification. The new classifier relaxes the assumptions on distance axioms and data independence. Furthermore, we consider each neighbor of the query instance as an information source providing a piece of evidence for reasoning on the target label. Then, Dempster's rule of combination is used to pool these pieces of evidence for making a decision.

The main contributions of this chapter are following:

- We introduce two new approaches that based on the mass estimation combining with the weighted sum method, and the theory of evidence respectively, for the imbalanced classification task.
- We measure the similarity between two instances based on mass,

instead of distance function that has weaknesses in the specific situation of the varied density of data points.
- The experiments are conducted on a wide-ranging application domains, a varied imbalance rates, different number of instances and features.
- We compare the proposed approaches with the other 11 existing classification algorithms in a wide range of learning methods.

The remaining parts of this chapter are structured as follows. Section 4.2 states the class imbalance problem. Section 4.3 introduces two new approaches including the k-LMN, and the EMass. Section 4.4 and its subsections present and discuss the experimental studies conducting on 60 imbalanced datasets. Experimental results shows that the proposed approaches outperforms the existing competitive models in terms of F1 score, Brier score, receiver operating characteristic curve (ROC score), and precision-recall curve (PR score). Finally, section 4.5 draws the conclusion and future directions.

4.2 Class imbalance statement

The classification is a popular supervised learning problem in the data mining field, which predicts the class label for new unlabeled instances based on the observed labeled data. Let X is the dataset of n instances, $X = \{x_1, \ldots, x_n\}$, and the corresponding class labels of all instances in X are known as $Y = \{y_1, \ldots, y_n\}$. These labels belong to a finite set number of classes $\Omega = \{l_1, l_2, ..., l_M\}$. The target of classification is to predict the most likely label $l_i \in \Omega$ for an unlabeled instance $\hat{x}$. Most of the classification methods use the similarities between instance $\hat{x}$ and all or several instances in X, which makes the similarity definition is essential in this research field.

A dataset is technically imbalanced when it has an unequal class distribution. For example, if there is a significant disproportion among the number of instances belonging to each class, then the given dataset is called imbalance. Hence, there may exist an underrepresented class label in the dataset. Therefore, classifying on an imbalanced dataset becomes more difficult because several classes can dominate the other ones in the total number of instances. That makes similarity standards are different for a set of classes where the query instance belongs. Hence, the basic similarity measures become no longer suitable in this case.

There is a real-world example that developing a medical application for differentiating between benign and malign tumors of a specific type of cancer using different features. In this case, it is much more essential to identify malign tumors correctly than benign ones. Because the consequence of

undetected malign tumors can be fatal, but a false positive to detect the tumor as benign might not be harmful. However, the conventional classifiers tend to have high accuracy for the majority class while achieving poor results for the minority class.

As a result, the instances belonging to the minority class are high frequently misclassified than those from the majority class. Hence, the accuracy metric is no longer a proper measure in the class imbalance. The other more informative metrics, such as ROC, F-series measure, Brier score, precision, and recall, are required to distinguish between the numbers of correctly classified instances of different classes.

In summary, we need to construct classifiers that are biased toward the minority class. Hence, this chapter shows how this goal can be achieved in order to build meaningful classifiers for the imbalanced datasets.

4.3 Methodology

4.3.1 Confidence estimation

The confidence of an instance x_i, or $conf(x_i)$ for short, represents how much confidence this instance belongs to class y_i. It can be calculated by the following Equation 4.1, regarding to the Bayes' theorem.

$$conf(x_i) = P(y_i|x_i) = \frac{P(y_i) \times P(x_i|y_i)}{\sum_{j=1}^{M} P(y_j) \times P(x_i|y_j)} \tag{4.1}$$

Where $y_i \in \Omega$, that $\Omega = \{l_1, l_2, ..., l_M\}$ is a finite set of M class labels. $P(y_j)$ is the prior probability of y_j, and $P(x_i|y_j)$ is the likelihood probability. To calculate this likelihood, we use the Gaussian mixture model for estimating class-wise probability density function.

4.3.2 Mass-based similarity measurement

The mass-based dissimilarity measurement introduced in [32] can be applied to compute the similarity between two instances. As introduced in chapter 2, on one side the similarity between two instances ($x_i, x_j \in X$) can be maximum when (x_i, x_j) are in the same leaf node of the hierarchical partitioning structure. On the other side, the similarity will be minimum when the two data points are in the root node.

To measure this similarity, normalization is applied as Equation 4.2, so that $sim(x_i, x_j) \in [0, 1 - \frac{2}{N}]$, where N is the number of instances, and

$mass_{max}$ is the maximum value of the estimated $mass_e(x_i, x_j)$ between two instances $(x_i, x_j), 1 \leq j \leq k$.

$$sim(x_i, x_j) = 1 - \frac{mass_e(x_i, x_j)}{mass_{max}} \tag{4.2}$$

4.3.3 Mass-based similarity weighted *k*-neighbor S*k*-LMN approach

(As of May 2021, the Sk-LMN approach is accepted to present at the 18th International Conference on Modeling Decisions for Artificial Intelligence MDAI 2021).

In this section, Sk-LMN approach is introduced for the class imbalance problem. The main idea in this approach is based on mass estimations instead of the distance-based functions to measure the dissimilarity between two instances. For this reason, Sk-LMN can overcome the key shortcomings of the distance-based or density-based classifiers.

Another key point is as uncertainty often exists in almost all datasets, the confidence of an instance plays an important role in the class imbalance task where a few information is available for the minority classes. This confidence represents a conditional probability (Equation 4.1) as the likelihood of a class label to which the query instance belongs. There are several methods that also compute the conditional probability for classifying an instance, e.g. NB classifier.

However, these methods cannot perform well for the imbalanced classification due to the weak estimation of the conditional density of the new instance associated with each class. Noticeably, Sk-LMN computes the conditional probability of neighbor instances belonging to the context set of the query instance rather than itself.

Next, a simple weighted sum is used to aggregate the weighted confidence values provided by each individual neighbor of the query instance. The flowchart of the proposed approach is graphically illustrated in Figure 4.1.

Specifically, for a new instance x_t, we find k lowest $mass_e$ neighbors (k-LMN) around it using the mass-based dissimilarity measurement as it was introduced in [32]. Let k-LMN(x_t) be a context set of the query instance x_t. Each member of the k-LMN(x_t), called x_i, assigns a weighted confidence value, which is computed by Equation 4.3), supporting to predict the class label of x_t.

We observed that firstly a neighborhood instance will provide more importance or larger weighted confidence value to class $y_j (1 \leq j \leq M)$ when this neighbor has higher confidence that it belongs to y_j. A neighbor with a

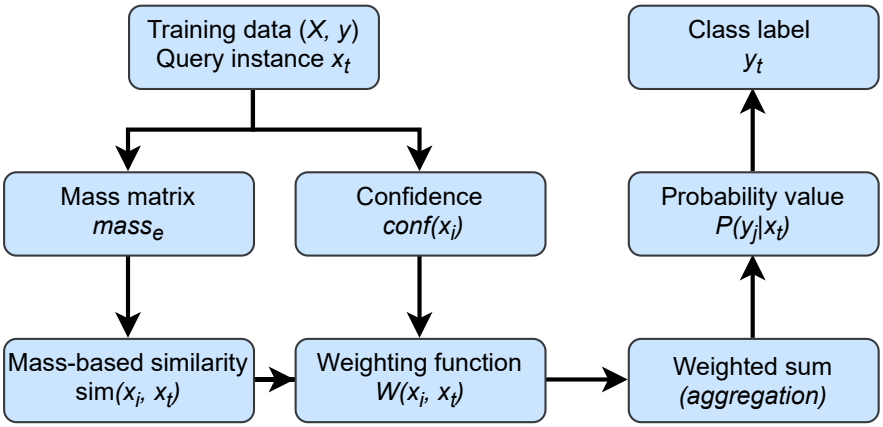


Figure 4.1: Flowchart of the Sk-LMN approach.

greater posterior probability should have larger confidence than the one which is in the lower posterior probability area. Secondly, a neighbor will calculate more importance or larger weighted confidence value to a specific class when the neighbor and the query instance have more similarity. We then formulate the weighting function that satisfies these two aforementioned observations as in Equation 4.3.

$$W(x_i, x_t) = sim(x_i, x_t) \times conf(x_i) \tag{4.3}$$

Where $conf(x_i)$ is the confidence of x_i represented by the posterior probability of class label y_i given x_i, and $sim(x_i, x_t)$ represents the mass-based similarity between x_i and x_t.

Algorithm 4.1 *train*Model(X, Y)

Input: training data (X, Y)
Output: $conf, mass_{max}$

Initialize array $conf$
$mass_{max} \leftarrow 0$
for $i = 1$ to cardinality of X **do**
  $conf[i] \leftarrow$ calculate confidence, //refer to Equation 4.1
  for $j = i + 1$ to cardinality of X **do**
    $mass \leftarrow mass_e(x_i, x_j)$
    $mass_{max} \leftarrow \max(mass, mass_{max})$
  end for
end for
return $conf, mass_{max}$

Algorithm 4.2 Sk-LMN pseudo code

Input: training data (X, Y), neighbor size k, query instance x_t
Output: class label y_t

$conf(x_i), mass_{max} \leftarrow train$Model$(X, y)$, from Algorithm 4.1
$s \leftarrow$ indices of k-LMN(x_t)
Initialize a list of W values
for $i = 1$ *to* k **do**
  $index \leftarrow s[i]$
  $confidence \leftarrow conf(x_{index})$
  $mass \leftarrow mass_e(x_t, x_{index})$
  $similarity \leftarrow$ using Equation 4.2
  $weight \leftarrow$ using Equation 4.3
end for
Combine *weighted confidence* values using Equation 4.4
$\hat{y}_t \leftarrow$ predict class label, //refer to Equation 4.5
return class label $\hat{y}_t$

4.3.3.1 Weighted sum aggregation

Assume further that, for every neighbor instance x_i in the context set k-LMN(x_t), x_i assigns a numerical weighted confidence value $W(x_i, x_t)$ to support class y_j as its relative importance to the query instance x_t. The weighted sum, which is probably the best known and widely-used method for calculating the comprehensive evaluation, is applied to score the total support in x_t. That is, for any query instance x_t we can compute the probability of

x_t is assigned to class $y_j (1 \leq j \leq M)$ as follows,

$$P(y_j|x_t) = \frac{\sum_{x_i \in k\text{-LMN}(x_t), y_i = y_j} sim(x_t, x_i) \times conf(x_i)}{\sum_{1 \leq l \leq M} \sum_{x_i \in k\text{-LMN}(x_t), y_i = y_l} sim(x_t, x_i) \times conf(x_i)} \tag{4.4}$$

4.3.3.2 Label prediction in Sk-LMN approach

It is important to realize that Equation 4.3 will return a larger weighted confidence value when a neighbor assigns more confidence and has more similarity to the query instance. To classify x_t, the weighted sum aggregation operator is applied as in Equation 4.4 to pool these discounted confidence values for each singleton class. According to this probability, we make the final decision by using Equation 4.5.

$$\hat{y}_t = \arg\max_{1 \leq j \leq M} P(y_j|x_t) \tag{4.5}$$

4.3.4 Mass-based similarity integrated with evidential reasoning: EMass approach

(The study on EMass approach will be submitted to the Journal of Information Fusion, in August 2021.)

The distance-based or density-based classifiers such as DT, k-NN, RF, SVM, Bagging and AdaBoost, have been challenging on the skewed class distribution. These approaches treat all instances the same, although most of these instances belong to the majority class. Then, misclassification may occur due to the selection of features that is not suitable for the class imbalance task.

In this work, we propose an EMass approach that provides more property for instances that belong to the minority class. As a result, the EMass can handle the misclassification issue for class imbalance tasks. In the EMass approach, the confidence of an instance $conf(x_i)$ represents a posterior probability knowing the prior probability and the likelihood of a class label that instance x_i belongs. $Conf(x_i)$ is calculated by Bayes's theorem on training data as in Equation 4.1. Next, the $conf(x_i)$ is weighted by the mass-based similarity measurement between the instance and its query as in Equation 4.3.

There are several classification methods, such as NB also computes the conditional probability for classifying an instance. However, this method cannot perform well for the class imbalanced problem because the weak

estimation of the conditioning density of the new instance associated with each class. On the other hand, the EMass approach computes the conditional probability of neighborhood instances belonging to the context set of the query instance rather than itself.

In addition, as uncertainty exists in almost all datasets, then the degree of confidence plays an important role in the imbalanced classification problems. To address this issue, Dempster's rule of combination is applied to combine the pieces of evidence provided by each individual neighbor of the query instance. The flowchart of the proposed approach is graphically illustrated in Figure 4.2.

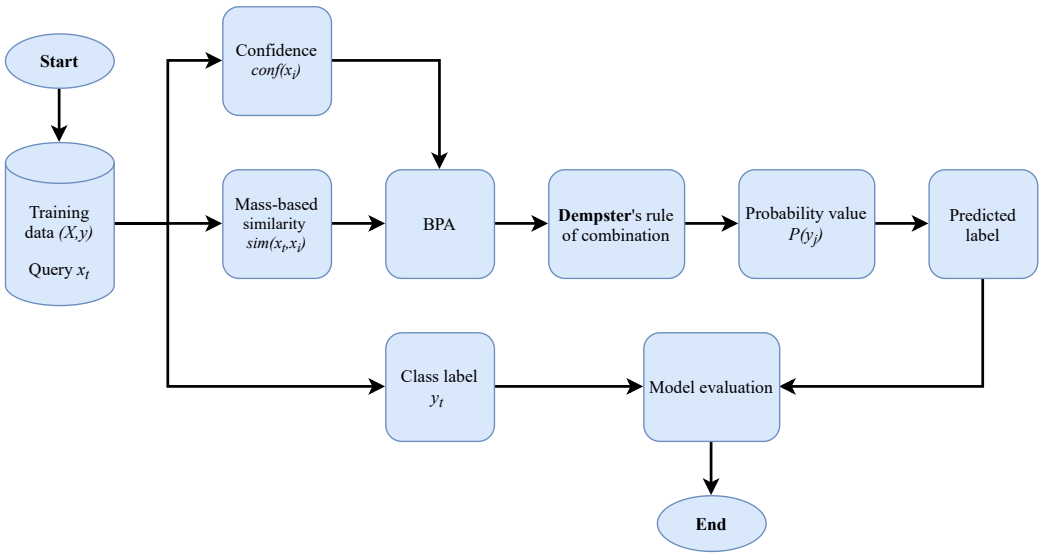


Figure 4.2: Flowchart of the EMass approach.

For a new query instance x_t, a context set (refer to Section 2.2.3 for more details) k-LMN(x_t) is obtained by gathering k-neighbors around x_t with the lowest mass-based similarities measurement as it is introduced in [32]. Each member of the k-LMN(x_t) is considered as an information source providing a piece of evidence, which assigns the basic probability (BPA) values for each subset of the set Ω, supporting to predict the class label of x_t.

We consider the i-th neighbor of the query instance x_t, so-called x_i, as an information source that provides a piece of evidence supporting to predict the class label $y_l, 1 \leq l \leq M$, by some belief. Thus, the remaining of this

belief cannot be committed to any other subset of the set Y except itself. The BPA values assigned by x_i can be represented by Equation 4.6, 4.7, and 4.8. Where $W(x_i, x_t)$ is the weighted confidence computed by Equation 4.3, and β is a hyperparameter, $0 < \beta < 1$.

$$m_i(\{y_l\}) = \beta \times W(x_i, x_t) \tag{4.6}$$

$$m_i(Y) = 1 - \beta \times W(x_i, x_t) \tag{4.7}$$

$$m_i(A) = 0, \quad \forall A \in 2^Y \setminus \{Y, \{y_l\}\} \tag{4.8}$$

It is interesting to observe that firstly a piece of evidence will assign a higher degree of belief or BPA value to class y_l when the evidence has more confidence that it belongs to y_l. In other words, a piece of evidence having a higher posterior probability should have more belief than the one which is in the lower posterior probability region. Secondly, a neighbor will assign more degree of belief or BPA value to a specific class when this neighbor and the query instance have more similarities.

Algorithm 4.3 EMass pseudo code

Input: training data (X, Y), neighbor size k, query instance x_t
Output: class label $\hat{y_t'}$

$conf, mass_{max} \leftarrow train$Model$(X, Y)$, //refer to Algorithm 4.1
$s \leftarrow$ indices of k-LMN(x_t)
Initialize a list of *weighted* values
for *i = 1 to k* **do**
    $index \leftarrow s[i]$
    $conf \leftarrow conf(x_{index})$
    $mass \leftarrow mass_e(x_t, x_{index})$
    $sim \leftarrow$ using Equation 4.2
    $BPA[x_i] \leftarrow$ using Equation 4.6, 4.7, 4.8
end for
Combine BPA values using Equation 2.5
$P(c) \leftarrow$ compute probability values, //refer to Equation 4.10
$\hat{y_t'} \leftarrow$ predict class label, //refer to Equation 4.11
return class label $\hat{y_t'}$

4.3.4.1 Dempster's rule of combination

The BPA values are assigned by each neighbor instance x_i, $x_i \in k - \text{LMN}(x_t)$, according to Equation 4.6, 4.7, 4.8. Then, the Dempster's rule of combination is applied to aggregate k pieces of evidence that support to predict whether each singleton class label y_l that the query instance x_t belongs.

$$m_t(\{y_l\}) = \oplus m_i(\{y_l\}), \quad i = 1, 2, ..., k \quad \text{and} \quad q = 1, 2, ..., M \tag{4.9}$$

Note that the Equation 4.9 is a general version of the Equation 2.5.

4.3.4.2 Label prediction in EMass approach

For making decision, the combined BPA values are converted into probability values. For each singleton class y_l, $q = 1, 2, ..., M$, $P_t(y_l)$ is derived by Equation 4.10.

$$P_t(y_l) = \sum_{y_l \subseteq B} \frac{|y_l \cap B|}{|y_l|} \times m_t(y_l) \tag{4.10}$$

Where B is a subset of the set Ω.

According to this probability, we make final decision using Equation 4.11.

$$\hat{y_t'} = \arg\max_{y_l \in \Omega} P_t(y_l) \tag{4.11}$$

Where y_l is a singleton class so that the cardinality of y_l is 1.

4.4 Experimental studies

The experimental studies were conducted on 60 imbalanced datasets to compare the performances of two proposed approaches, Sk-LMN and EMass respectively, with the other 11 competitive methods. Then, the Wilcoxon signed ranks test is employed as a non-parametric statistical analysis to validate those experimental results.

4.4.1 Dataset description

The imbalanced datasets were collected from the knowledge extraction based on the evolutionary learning (KEEL) [59], and UCI repository [92] to conduct experiments on wide-ranging application domains, different numbers

of instances, numbers of features, and a variety of imbalance ratios. The imbalance ratio (IR) between the samples of the majority class and minority class of the datasets used in these experiments are from 1.82 to 100.14. A dataset is higher imbalanced when the value of IR is bigger. These datasets have prepared for class imbalance tasks as shown in Table 4.1, which summarizes the characteristics for 60 imbalanced datasets.

Table 4.1: Descriptions of 60 imbalanced datasets. Idx., #Inst., #Ftr., and IR represent index of dataset, number of instances, features, and imbalance rate respectively.

Idx.	Dataset	#Inst.	#Ftr.	IR	Idx.	Dataset	#Inst.	#Ftr.	IR
1	Glass1	214	9	1.82	31	Glass-0-1-4-6_vs_2	205	9	11.81
2	Ecoli-0_vs_1	220	7	1.89	32	Glass-0-6_vs_5	108	9	12.50
3	Iris0	150	4	2.06	33	Ecoli-0-1-4-6_vs_5	280	6	13.74
4	Glass0	214	9	2.10	34	Shuttle-c0-vs-c4	1829	9	13.87
5	Haberman	306	3	2.78	35	Glass4	214	9	16.83
6	Vehicle2	846	18	2.88	36	Dermatology-6	358	34	16.90
7	Vehicle1	846	18	2.90	37	Winequality-white-9_vs_4	168	11	17.67
8	Vehicle3	846	18	2.99	38	Ecoli4	336	7	17.68
9	Vehicle0	846	18	3.25	39	Zoo-3	101	16	19.20
10	Ecoli1	336	7	3.36	40	Poker-9_vs_7	244	10	19.50
11	New-thyroid1	215	5	5.14	41	Shuttle-c2-vs-c4	129	9	20.50
12	Newthyroid2	215	5	5.32	42	Glass-0-1-6_vs_5	184	9	22.00
13	Segment0	2308	19	6.02	43	Shuttle-6_vs_2-3	230	9	22.00
14	Glass6	214	9	6.38	44	Glass5	214	9	25.75
15	Yeast3	1484	8	8.10	45	Winequality-red-4	1599	11	29.17
16	Ecoli3	336	7	8.60	46	Kddcup-guess_passwd_vs_satan	1642	38	29.98
17	Page-blocks0	5472	10	8.79	47	Yeast-1-2-8-9_vs_7	947	8	31.66
18	Yeast-0-3-5-9_vs_7-8	506	8	9.12	48	Abalone-3_vs_11	502	7	32.47
19	Yeast-0-2-5-7-9_vs_3-6-8	1004	8	9.14	49	Ecoli-0-1-3-7_vs_2-6	281	7	39.42
20	Ecoli-0-3-4_vs_5	200	7	9.53	50	Abalone-21_vs_8	581	7	40.50
21	Ecoli-0-6-7_vs_3-5	222	7	9.57	51	Yeast6	1484	8	41.40
22	Ecoli-0-1_vs_2-3-5	244	7	9.61	52	Kddcup-land_vs_portsweep	1061	38	49.52
23	Ecoli-0-2-3-4_vs_5	202	7	9.63	53	Abalone-19_vs_10-11-12-13	1622	7	49.69
24	Ecoli-0-2-6-7_vs_3-5	224	7	9.67	54	Poker-8-9_vs_6	1485	10	58.40
25	Ecoli-0-4-6_vs_5	203	6	9.68	55	Shuttle-2_vs_5	3316	9	66.67
26	Vowel0	988	10	9.98	56	Kddcup-buffer_overflow_vs_back	2233	38	73.43
27	Glass-0-1-6_vs_2	192	9	10.29	57	Kddcup-land_vs_satan	1610	38	75.67
28	Glass-0-4_vs_5	92	9	10.50	58	Poker-8-9_vs_5	2075	10	82.00
29	Ecoli-0-6-7_vs_5	220	6	10.58	59	Poker-8_vs_6	1477	10	85.88
30	Led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	11.31	60	Kddcup-rootkit-imap_vs_back	2225	38	100.14

These datasets were originally collected for the conventional binary or multiple classification tasks. For binary class imbalanced problems, the original datasets were prepared by considering one of the minority class(es) as a positive class (class 1), and the remaining class(es) were processed to form the negative class (class 0). We used the following original classification datasets to prepare the imbalanced datasets as presented in Table 4.1.

First, "Glass" dataset used to experiment with the class imbalance approaches. The collected glasses, which were presented by nine attributes corresponding to the glass types. These glasses were considered as evidences by the investigators. In total, there are 214 instances separated in all glass types. "Glass1" was prepared that the glass type 1 formed the positive class and all the other glass types as the negative class. In the same manner, we

prepared "Glass0", "Glass4", "Glass6", and "Glass-0-4_vs_5".

Second, "Wisconsin-Breast Cancer", or "Wiscosin" for short, including 683 medicine diagnoses were collected with 9-dimensions features. The malignant class was considered as positive class and the benign as negative class. The IR is 1.86.

Third, "Pima" data was first gathered by the National Institute of Diabetes, Digestive, and Kidney Diseases to predict whether a person was likely to have diabetes. All joined people were females, and they were 21 years old by that time. There are 768 recorded instances represented in 8-dimensions within 268 positive instances (IR = 1.87) in the Pima set.

Continuously, "Ecoli" set could be downloaded from the UCI website. Seven numerical attributes were used to form the positive class and negative class, and the "sequence name" attribute were eliminated from the "Ecoli" set. The prepared Ecoli imbalanced datasets resulted in the following name: "Ecoli-0_vs_1", "Ecoli1", "Ecoli2", "Ecoli-0-6-7_vs_3-5", "Ecoli-0-1_vs_2-3-5", "Ecoli-0-2-3-4_vs_5", "Ecoli-0-2-6-7_vs_3-5", "Ecoli-0-4-6_vs_5", "Ecoli-0-3-4-7_vs_5-6", "Ecoli-0-3-4-6_vs_5", "Ecoli-0-6-7_vs_5", "Ecoli-0-6-7_vs_3-5", "Ecoli-0-1-4-7_vs_2-3-5-6", "Ecoli-0-1_vs_5", "Ecoli-0-1-4-7_vs_5-6", "Ecoli-0-4-6_vs_5".

Then, "Vehicle" or vehicle silhouettes [93] data was original collected at the TI in 1986-87 by JP Siebert. The purpose is to classify a given silhouette as one of four types of vehicles, using a set of attributes extracted from the hierarchical image processing system (HIPS). This system obtains a combination of scale-independent features from scaled variance and skewness about the major/minor axes as classical moments-based measures. In this experiment, we tested all 12 competitive models on vehicle1 and vehicle2 datasets.

Next, "New-thyroid" dataset was downloaded from the UCI website. The instances in this dataset are labeled with the normal, subnormal functioning, and hyperfunction by 15 nominal attributes and six numerical ones. The New-thyroid set is employed to classify whether a patient indicated to the clinic was hypothyroid. In this work, the subnormal class and hyperfunction classes together form a positive class. The normal one forms a negative class. Both classes have five numeric features.

"Segnment" or image segmentation dataset was created by vision group, University of Massachusetts. In this dataset, the instances were drawn randomly from a database of seven outdoor images. The images were manually segmented to create a classification task for every pixel. Each instance is a 3x3 region. Segment0 has experimented for this class imbalance experiment with 2308 instances represented by 19 features.

"Yeast" dataset [94] contains information about a set of Yeast cells. The

original task is to determine the localization site of protein among ten possible alternatives. For this class imbalance task, both classes were formed by picking suitable class(es) from 10 classes under the imbalance ratio values. For example, in "Yeas-2_vs_4" set, class-2 is considered as the positive class and class-4 as the negative class.

"Page-blocks" dataset contains blocks of the page layout of a document that has been detected by a segmentation process. This dataset has 5472 instances that came from 54 different documents. Each observation concerns one block. The original task is to determine the type of block that includes text (0), horizontal line (1), graphic (2), vertical line (3), or picture (4). We prepared "Page-blocks0" and "Page-blocks-1-3_vs_4" for the comparison among 12 classifiers.

"Vowel" recognition dataset contains information about speaker-independent recognition of the eleven reliable state vowels of British English using a specific training dataset that derived log area ratios. This version is a merge of the two original datasets present at the UCI repository.

"Led7digit" dataset contains 7 Boolean attributes, one for each light-emitting diode of a 7-segment display representing ten classes, the set of decimal digits. The original task is to determine which digit has been showing on the screen. There are noises introduced in these datasets. Therefore, each attribute value has a different probability. In this experiment, class-1 (represented number 1) was treated as a negative class, and the remaining classes formed the positive class.

All "Shuttle" datasets here have the same nine dimensions but different numbers of instances. The first "Shuttle-0_vs_4" set has 1829 instances that class-0 as belonging to the positive class, and class-4 forming the negative class. The second "Shuttle-2_vs_4" set contains 129 instances, where class-2 was taken as positive class and class-4 as negative class. The third "Shuttle-6_vs_2-3" set consists of 230 instances, class-6 formed the positive class, class-2 and class-3 together formed the negative class, and the remaining classes were deleted. The fourth "Shuttle-2_vs_5" set has 3316 instances, class-2 was considered as positive class and class-5 as a negative one.

"Dermatology" dataset contains 34 attributes, 33 of which are linear values and one of them is nominal. In this dataset, the differential diagnosis of an erythema-squamous disease is a real challenge. They all share the clinical features of erythema and scaling, with very few differences. Patients were first evaluated clinically with 12 attributes. Afterward, skin samples were collected for the evaluation of 22 histopathological features.

"Winequality" datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical and sensory variables are available. The classes of this

dataset are ordered and not balanced because there are many more normal wines than excellent or poor ones. The 12 competitive algorithms are employed to predict excellent or poor wines.

"Zoo" dataset contains 16 Boolean-valued attributes. The "type" attribute appears to be the class attribute. The original task here is to classify which animals are in which type.

In "Pocker" set, each record is an example of a hand consisting of five playing cards drawn from a standard deck of 52 cards. Each card is represented by two attributes. There is one class attribute that describes the "Pocker Hand". Note that the order of the card is important.

Next, the KDDCup99 dataset has 34-continuous and 7 nominal features. The numbers of feature were downed to four ones: duration, dst-bytes, src-bytes, and service. Using the nominal feature 'service', the data were separated into HTTP, SMTP, FTP, FTP-data, and other subsets. Then, the classification tasks were experimented on each subset.

Finally, the abalone dataset represents physical measurements for determining the age of abalone. This dataset can be used by a machine-learning algorithm to predict the ages of a new abalone. In this study, we considered each individual sex class as a positive class with slightly different fractions.

4.4.2 Implementation details and evaluation metrics

Both proposed approaches, Sk-LMN and EMass, were compared with other 11 competitive methods. These methods include the conventional learning algorithms (C4.5 DT, NB, k-NN), logistic regression (LR), tree-based recent algorithms for imbalanced classification (RF), linear support vector machine (LinearSVM), SVM with RBF kernel (RBF_SVM), ensemble learning (Bagging, AdaBoost, and XGBoost), and recent evidential algorithm (mPEkNN).

There are many methods and aspects to evaluate the performance of a system for class imbalance problems, e.g. time, space, accuracy rate, F-series score, G-mean, Brier score, and the area under the curve (AUC) values. However, we consider the area under the precision-recall curves (PR-AUC values) as the most important factor due to its popularity in the literature. Moreover, we also include the F1 score of all testing models as well. These two metrics are used to assess the performance of the 12 competitive models.

Besides, most of the classifiers have demonstrated beyond the binary classification as a multi-class problem that can simplify by the two-class task. Regularly, the minority class label is positive (or 1), and the majority class label is negative (or 0). In that case, the outcome of a classifier has been

represented by a confusion matrix. This matrix has been used to calculate the F1 score and AUC values.

In addition, the Brier score computes the mean squared error between predicted probabilities and the ground truth values. This score summarizes the magnitude of the error in the probability forecasts ans is suitable for binary classification problems. Brier score might be an appropriate probabilistic metric for class imbalance problems when it focus on evaluating the probabilities for the positive class. The error score is always between 0.0 and 1.0, where a model with perfect skill has a score of 0.0.

Finally, we have conducted the tenfold cross-validation test to evaluate the performance of 12 tested methods. As a result, these classifiers have ranked on each dataset in terms of the F1 score, and PR-AUC value, where a lower-ranked number or a higher average value indicates better performance.

4.4.3 Results and discussions

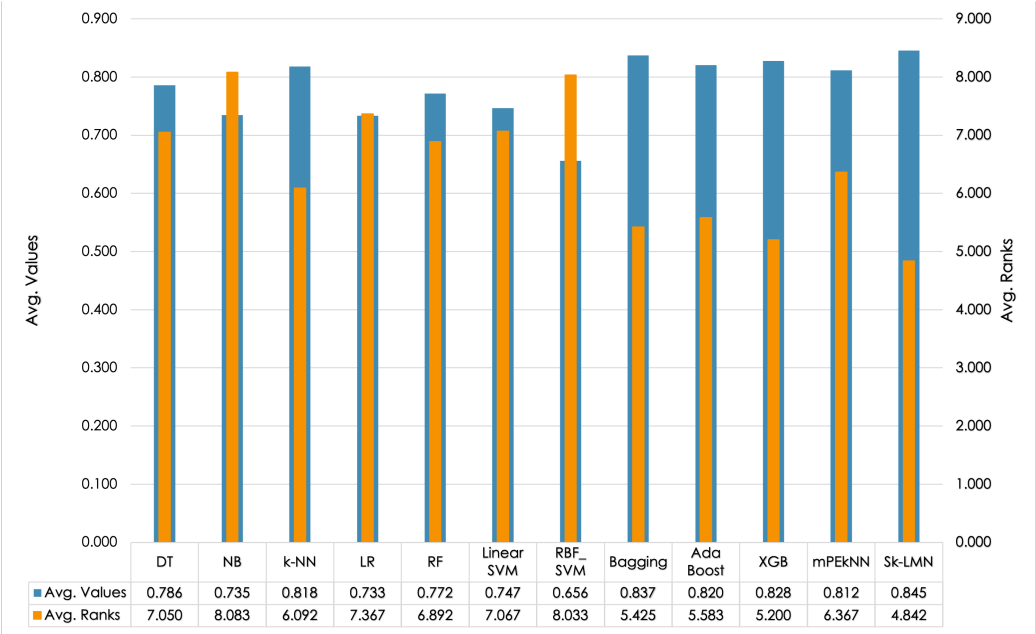
4.4.3.1 Sk-LMN approach

As can be seen in Figure 4.3, we compared the 12 tested models on the PR-AUC evaluation metric from the tenfold cross-validation executed on the 60 real imbalanced datasets. It is worth noticing that the Sk-LMN model outperforms all the other tested models in both average values (**0.845**) and average ranks (**4.842**).

Figure 4.4 shows the F1 score comparison results of the 12 models on the same tested datasets. These results show that the Sk-LMN approach achieved the best average value (**0.738**) in the F1 score metric. However, the Sk-LMN model has reached the second-best average rank (**5.250**) while the best one (**4.808**) is the XGB method.

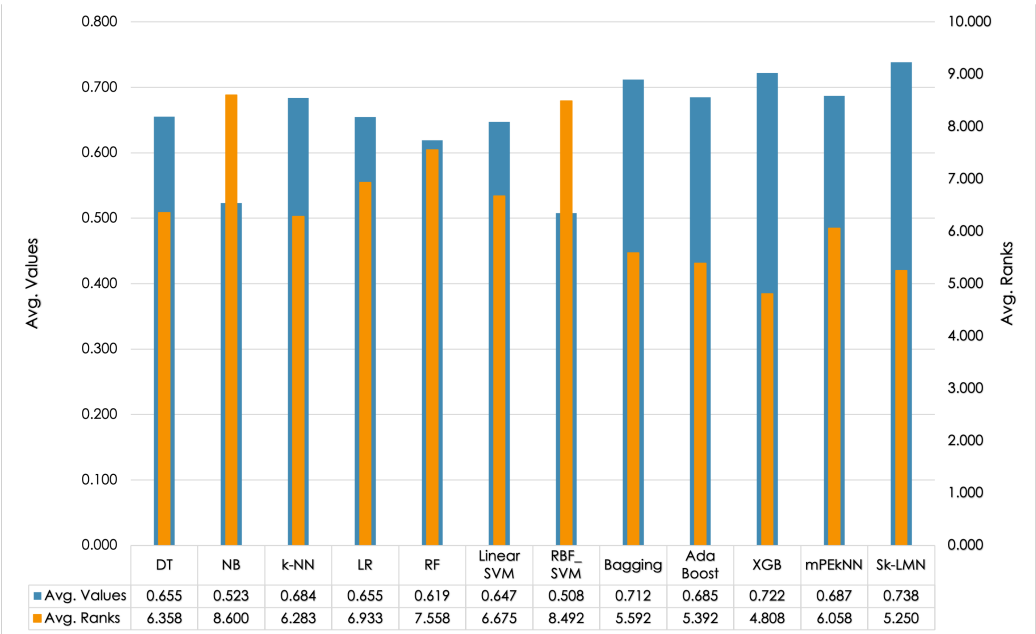
4.4.3.2 Non-parametric statistical test results

The Wilcoxon signed ranks test [35] was sellected as a non-parametric statistic test to accomplish pairwise comparisons among the Sk-LMN approach and the others. These test aim to validate all the experimental results. The Wilcoxon analysis states the sum of the signed-ranks for each comparing pairwise methods. R^+ is denoted as the sum of the positive rank values and R^- as the sum of the negative rank values. The IBM SPSS software has been employing with experimental results tested on the 60 imbalanced datasets.



	DT	NB	k-NN	LR	RF	Linear SVM	RBF_ SVM	Bagging	Ada Boost	XGB	mPEkNN	Sk-LMN
Avg. Values	0.786	0.735	0.818	0.733	0.772	0.747	0.656	0.837	0.820	0.828	0.812	0.845
Avg. Ranks	7.050	8.083	6.092	7.367	6.892	7.067	8.033	5.425	5.583	5.200	6.367	4.842

Figure 4.3: Sk-LMN comparison results on PR-AUC results.



	DT	NB	k-NN	LR	RF	Linear SVM	RBF_ SVM	Bagging	Ada Boost	XGB	mPEkNN	Sk-LMN
Avg. Values	0.655	0.523	0.684	0.655	0.619	0.647	0.508	0.712	0.685	0.722	0.687	0.738
Avg. Ranks	6.358	8.600	6.283	6.933	7.558	6.675	8.492	5.592	5.392	4.808	6.058	5.250

Figure 4.4: Sk-LMN comparison results on F1 scores.

Table 4.2: Wilcoxon signed ranks test on Sk-LMN comparison results.

Sk-LMN vs.	F1 results			ROC-AUC results			PR-AUC results		
	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
DT	226.5	676.5	0.005	136.0	725.0	0.001	221.0	682.0	0.004
NB	92.0	898.0	0.001	171.0	732.0	0.001	126.0	820.0	0.001
k-NN	309.0	511.0	0.174	105.0	598.0	0.001	190.0	513.0	0.015
LR	260.5	820.5	0.002	288.0	573.0	0.065	121.0	782.0	0.001
RF	132.5	857.5	0.001	284.0	662.0	0.022	178.5	811.5	0.001
LinearSVM	264.0	771.0	0.004	294.0	447.0	0.267	156.0	705.0	0.001
RBF_SVM	115.5	1262.5	0.001	199.0	791.0	0.001	159.0	876.0	0.001
Bagging	357.5	503.5	0.344	357.0	346.0	0.934	292.0	411.0	0.369
AdaBoost	294.0	447.0	0.267	371.0	409.0	0.791	321.5	458.5	0.339
XGB	470.0	476.0	0.971	450.0	370.0	0.591	391.5	496.5	0.613
mPEkNN	230.5	472.5	0.086	90.0	576.0	0.001	149.0	517.0	0.004

Table 4.2 reports the Wilcoxon signed ranks test results for the F1 score and AUC metrics to perform multiple pairwise comparisons among the Sk-LMN and the other methods. It can be seen that RBF_SVM method achieved the best R^- score compared with the other testing models, but the proposed approach outperforms the RBF_SVM, according to the results of multiple comparisons. There are significant differences between the two methods with a confidence level higher than 99.9% (p-value = 0.001).

4.4.3.3 EMass approach

Figure 4.5 shows the average F1 score comparison for 12 models from the tenfold cross-validation test on 60 imbalanced datasets. Overall, the EMass approach achieved the best results on both the average values (**0.857**) and average ranks (**7.100**). Table 5.1 presents the detailed F1 score comparison.

Figure 4.6 illustrates the comparisons among the 12 tested models in terms of the average Brier score from the tenfold cross-validation test on the 60 imbalanced datasets. It can be seen that the proposed model outperformed all the other models in both average values (**0.034**) and average ranks (**9.067**). Note that a lower Brier score indicates better performance, and the k-NN model also obtained the best average values on the Brier score comparison. The detailed Brier scores are reported in Table 5.2

Figure 4.7 presents the comparison of the ROC-AUC results for the 12 tested models on the 60 imbalanced datasets. The comparison result shows that the EMass model achieved the best average ROC-AUC values (**0.959**) while RBF_SVM model had the worst one (**0.800**). In terms of the average ranks, AdaBoost obtained the highest result (**6.283**) meaning that AdaBoost outperformed all the other competitive models. The detailed ROC-AUC

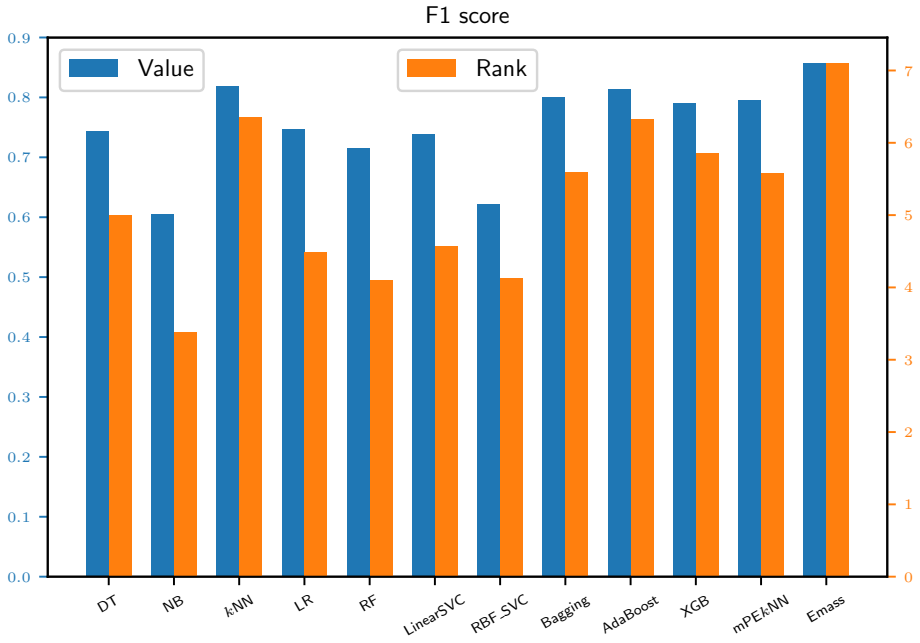


Figure 4.5: EMass comparison results on F1_score.

measurements are reported in Table 5.3.

Figure 4.8 shows the average PR-AUC results comparison among 12 models from tenfold cross-validation tested on the same datasets. Overall, the proposed approach outperforms all the other competitive models on both the average values (**0.915**) and average ranks (**6.183**). Table 5.4 reports the detailed PR-AUC results comparison.

4.4.3.4 Non-parametric statistical test results

The Wilcoxon signed ranks test [35] was used as a non-parametric statistical analysis to validate the experimental results in terms of the F1 score, Brier score, and AUC values. This analysis accomplishes multiple pairwise comparisons among the EMass approach and the other methods. The IBM SPSS statistics software has been employed on the experimental results and the output is presented in Table 4.3. Note that R^+ is defined as the sum of the positive signed-ranks and R^- as the sum of the negative signed-ranks for the results of each comparing method.

It can be seen from Table 4.3 that the RBF_SVM model achieved the best R^- results in terms of the F1 score (**1362.5**) and PR-AUC metric (**883.0**) compared with the other methods, but the proposed approach outperforms it according to the multiple methods comparison results presented in Table 5.1

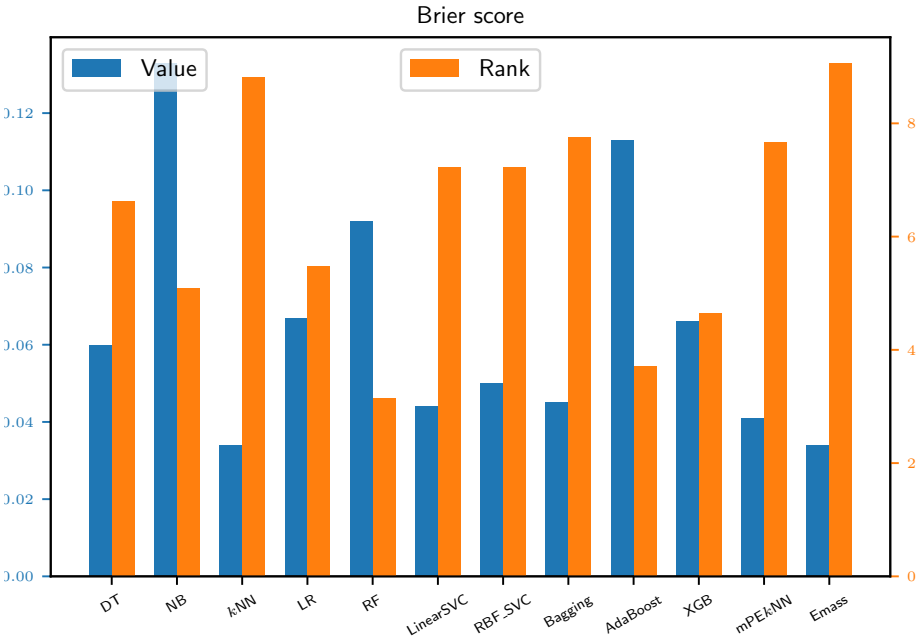


Figure 4.6: EMass comparison results on Brier_score.

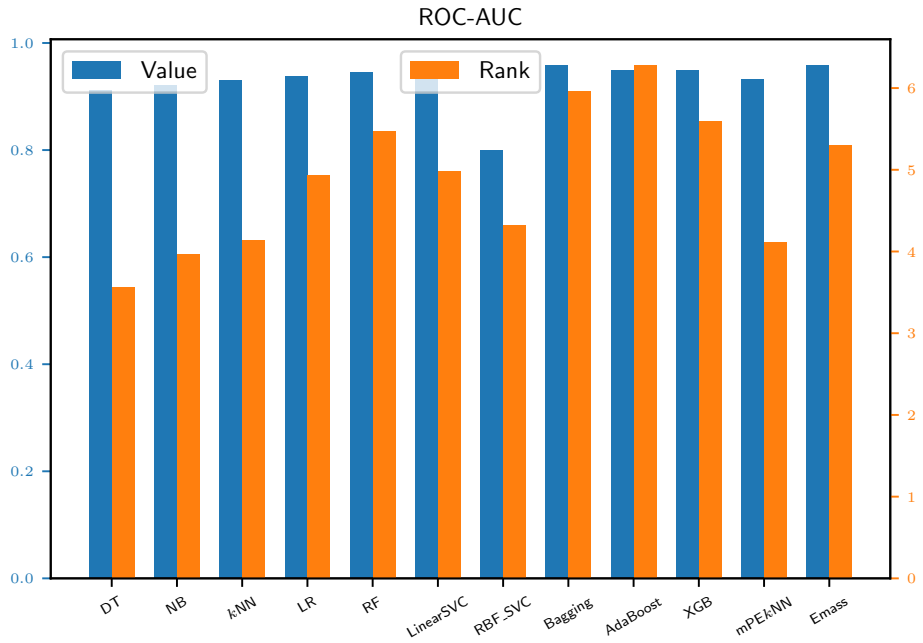


Figure 4.7: EMass comparison results on ROC-AUC.

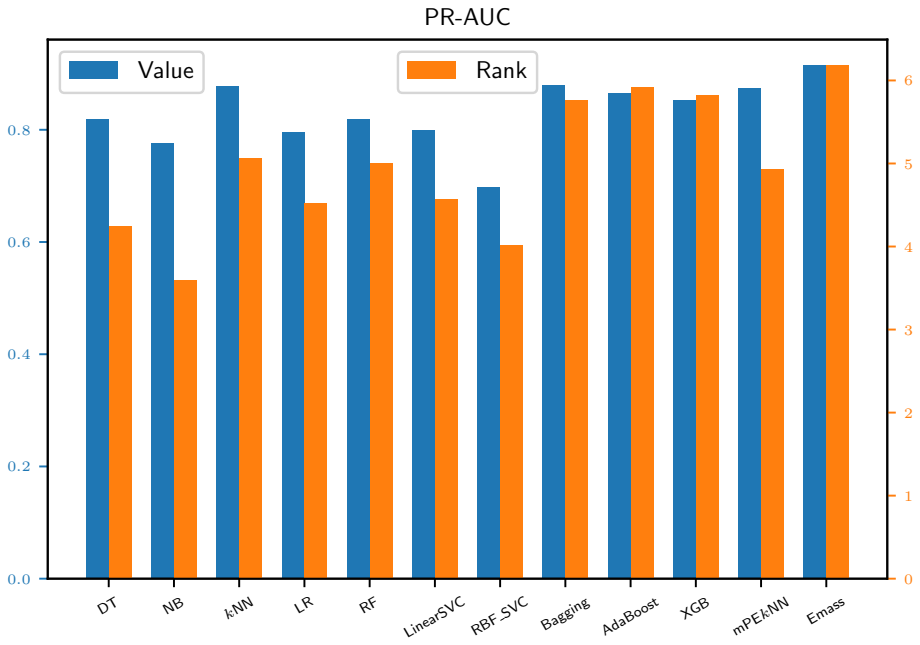


Figure 4.8: EMass comparison results on PR-AUC.

and Table 5.4. There are significant differences between the EMass and RBF_SVM with a confidence level higher than 99.9% (p-value = 0.001).

In terms of the Brier score, k-NN is the best algorithm from the pairwise comparison versus the EMass due to the highest R^- value (**758.0**), yet the EMass approach outperforms it regarding the results of multiple methods comparison in Table 5.2. On the other hand, RF is the worst model among the other methods. In the metric of ROC-AUC, DT model obtained the best R^- value (**772.0**) compared to the other algorithms and the test reports a p-value = 0.001. It means that the confidence level is higher than 99.9% for the pairwise methods comparison result.

Table 4.3: Wilcoxon signed ranks test on EMass comparison results.

EMass vs.	F1 results			Brier results			ROC-AUC results			PR-AUC results		
	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
DT	150.0	840.0	0.001	1075.0	200.0	0.001	174.0	772.0	0.001	156.0	790.0	0.001
NB	74.0	1054.0	0.001	1592.0	119.0	0.001	259.5	643.5	0.016	77.0	869.0	0.001
k-NN	256.5	373.5	0.338	727.0	758.0	0.894	106.0	455.0	0.002	191.5	438.5	0.043
LR	111.5	923.5	0.001	1572.0	258.0	0.001	349.5	511.5	0.294	152.0	751.0	0.001
RF	119.0	1057.0	0.001	1712.0	58.0	0.001	447.0	499.0	0.754	235.5	754.5	0.002
LinearSVM	120.0	915.0	0.001	1291.0	539.0	0.006	340.0	521.0	0.241	209.0	737.0	0.001
RBF_SVM	122.5	1362.5	0.001	1202.0	628.0	0.035	238.0	752.0	0.003	152.0	883.0	0.001
Bagging	228.0	592.0	0.014	1058.0	373.0	0.002	399.5	266.5	0.296	247.0	419.0	0.177
AdaBoost	206.5	496.5	0.029	1741.0	89.0	0.001	438.0	265.0	0.192	238.0	465.0	0.087
XGB	284.5	705.5	0.014	1664.0	166.0	0.001	400.5	460.5	0.697	290.0	571.0	0.069
mPEkNN	186.5	633.5	0.003	1027.0	458.0	0.014	114.5	480.5	0.002	184.5	481.5	0.020

4.5 Chapter conclusions

This chapter introduces two new classifiers for imbalanced datasets under perspectives of the mass-based measurement, neighbor-based algorithm, information fusion, and evidential reasoning approach. In both proposed approaches, the confidence of an instance is formulated as the posterior probability that measures the uncertainty of its class label. The Gaussian mixture model is applied to estimate the likelihood of the class label to compute the confidence. Then, the similarity between the query instance and its neighbor instance has been utilizing to weigh the estimated confidence.

The experimental study reveals that the weighted confidence method increases the likelihood of a minority class classification. In other words, the proposed approaches provides more importance to instances belonging to a positive (minority) class. The experiments conducted on 60 imbalanced datasets demonstrate that the proposed approaches outperforms the other 11 competitive methods on the PR-AUC evaluation metric for the Sk-LMN; and the F1 score, Brier score, AUC measures for the EMass approach.

This work highlights that the challenge of the misclassification issue can be handle when we refer directly to each neighbor instead of the query instance itself. Beside, the mass-based model has been exploiting to measure the similarity between two data points to address the shortcomings of the distance-based or density-based classifiers.

In addition, the previous classifiers tempt to rely on distance functions: this can be restrictive as it assumes that the datasets are independent and the distance axioms are satisfied. However, we have no such assumptions in the real-world applications. The data are dependent, uncertainty naturally, and the distance axioms failed if the data can not present in a geometric model. The EMass approach is quite different: it takes advantage of the mass-

based similarity measurement over distance-based functions for computing the similarity between two instances. Then, EMass follows Dempster's rule of combination to aggregated pieces of evidence for reasoning under uncertainty, while each neighbor is considered as an information source to provide the evidence.

However, we have to acknowledge the fact that both proposed approaches computes on the numeric variables only. In future works, we plan to extend the Sk-LMN and EMass approach for categorical features, mixing data, and experiment on more real-world datasets.

The source code and datasets of the Sk-LMN project have been organized and available on Github at the following link:
https://github.com/ImbOut/Sk-LMN.

Chapter 5

Conclusions and future works

Classification, outlier detection, and clustering techniques play significant roles in knowledge discovery and data mining (KDD), machine learning to better understand the data, recognize patterns, extract information, or create new knowledge. A vast of methods have been introduced in literature and applied in a wide range of domains. However, there is still having room for new researchers who conduct original experiments to contribute toward science by the systematic collection, interpretation, and evaluation of the data models.

By doing this research, we have answered the three main questions already. For each research objective, we firstly started by formulating the problem as clear and simple as possible. Therefore, the conceptual formulation question was answered firstly. Then, particularly, we introduced a new mass-based approach for local outlier detection, or MLOS for short. In addition, we continued to study imbalanced datasets and proposed the other two new methods for the class imbalance problems. The first model is called a mass-based similarity weighted k-neighbor for class imbalance, or Sk-LMN for short. This model combines the mass-based dissimilarity measurement with the weighted sum framework to built a new classifier. The second model is the other extension of the Sk-LMN model, the so-called EMass approach, in which Dempster-Shafer's theory of evidence was exploited. This method integrates the mass-based measurement with the evidential reasoning theory to handle the key shortcomings of distance-based or density-based classifiers. Hence, the second and third research questions were answered together while we proposed new models and conducted the experimental study on both the synthetic and benchmark datasets.

In summary, we firstly review the mass-based measurement in chapter 2. That is an alternative method to compute the dissimilarity between two data points. Distance-based functions such as Euclidean distance or Manhattan distance are popular in KDD and data science for measuring how data samples are related to each other. However, these distance-based measures have some weaknesses. Then, we reply to the distance-based functions by the mass-based measures to assess the relationship among instances.

In addition, we remind the main concepts of Dempster-Shafer's theory of evidence (DST). That attempt to assign the basic probability to each neighbor of the query instance then combine these pieces of evidence for making a decision. Next, the evaluation metrics are reviewed as a significant step for developing any machine learning model, which figures out the best model after training. The F1 score, Brier score, and AUC values are our main evaluation metrics for studying the imbalanced datasets. Finally, we utilized the Wilcoxon signed ranks test as a non-parametric statistical analysis to validate the experimental results.

In chapter 3, we formulate the outlier detection problem firstly. Then, the related works have been reviewing, and the shortcomings of the distance-based or density-based outlier models have been figured out. To address the shortcomings of the previous outlier detectors, we introduced a new mass-based approach for local outlier detection, the so-called MLOS approach. Then, the experiments are conducted on both synthetic datasets and benchmark datasets. The experimental results show that our proposed local outlier model works well on a wide-ranging of application domains, various numbers of feature and instance, and a varied imbalanced rate. The EMass approach can also be adjusted for novelty detection task to determine whether a query instance is an outlier.

In chapter 4, we describe the imbalance classification problem within its application domains. Then we propose two related models for solving the class imbalance issues. The first model is called Sk-LMN, in which the similarity between two data points is computed based on the mass estimation. Then, the simple weighted sum aggregates the information from the k-neighbor to make a decision. The second model is the extended version of the previous one, the so-called EMass. In this model, the Dempster-Shafer theory of evidence is utilized instead of using the weighted sum. Each neighbor of the query instance is considered as a piece of evidence. Then the Dempster's rule of combination is used to pool the evidence. The experimental study was conducted on 60 benchmark datasets with 12 tested competitive models. The results show that our two proposed models outperform the existing competitive models on several evaluation metrics such as F1 score, PR-AUC, and ROC-AUC values.

In conclusion, this work has contributed three new approaches to KDD and data science for outlier detection task and class imbalance problem respectively. The first model called MLOS approach for the local outlier detection task, and the other two models are called Sk-LMN and EMass for the class imbalance problems. We knowledge that there are limitations in this research. Then, we raise several future directions for the next plan as presented at the end of each chapter.

Publications

[1] A. Hoang, T. N. Mau, D. -V. Vo and V. -N. Huynh, "A Mass-Based Approach for Local Outlier Detection", *IEEE Access*, volume number: 9, page: 16448-16466, 2021.

[2] A. Hoang, T. N. Mau, V. -N. Huynh, "Mass-based Similarity Weighted k-Neighbor for Class Imbalance". *The 18th International Conference on Modeling Decisions for Artificial Intelligence.* Number of pages: 12. 27 – 30 September 2021. Umea, Sweden.

[3] C. Phan, A. Hoang, D. Phan, H. Dao and V. -N. Huynh, "Human Density Estimation by Exploiting Deep Spatial Contextual Information". *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ).* Page numbers: 1-5, 2 – 4 December 2019. Dunedin, New Zealand.

[4] D. -V. Vo, A. Hoang, and V. -N. Huynh, "An Evidential Reasoning Framework for User Profiling Using Short Texts." *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making.* Page numbers: 137-150, 11 – 13 November 2020. Phuket, Thailand.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[2] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2014.

[3] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[4] Y. Djenouri, A. Zimek, and M. Chiarandini, "Outlier detection in urban traffic flow distributions," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 935–940.

[5] R. Yu, X. He, and Y. Liu, "Glad: group anomaly detection in social media analysis," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 2, pp. 1–22, 2015.

[6] R. Yu, H. Qiu, Z. Wen, C. Lin, and Y. Liu, "A survey on social media anomaly detection," *ACM SIGKDD Explorations Newsletter*, vol. 18, no. 1, pp. 1–14, 2016.

[7] M. Riazi, O. Zaiane, T. Takeuchi, A. Maltais, J. Günther, and M. Lipsett, "Detecting the onset of machine failure using anomaly detection methods," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2019, pp. 3–12.

[8] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, "Anomalynet: An anomaly detection network for video surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2537–2550, 2019.

[9] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine learning*, vol. 30, no. 2, pp. 195–215, 1998.

[10] B. M. Haddad, S. Yang, L. J. Karam, J. Ye, N. S. Patel, and M. W. Braun, "Multifeature, sparse-based approach for defects detection and classification in semiconductor units," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 145–159, 2016.

[11] T. Lee, K. B. Lee, and C. O. Kim, "Performance of machine learning algorithms for class-imbalanced process fault detection problems," *IEEE Transactions on Semiconductor Manufacturing*, vol. 29, no. 4, pp. 436–445, 2016.

[12] P. Santos, J. Maudes, and A. Bustillo, "Identifying maximum imbalance in datasets for fault diagnosis of gearboxes," *Journal of Intelligent Manufacturing*, vol. 29, no. 2, pp. 333–351, 2018.

[13] H.-L. Dai, "Imbalanced protein data classification using ensemble ftm-svm," *IEEE transactions on nanobioscience*, vol. 14, no. 4, pp. 350–359, 2015.

[14] S. Lertampaiporn, C. Thammarongtham, C. Nukoolkit, B. Kaewkamnerdpong, and M. Ruengjitchatchawalya, "Heterogeneous ensemble approach with discriminative features and modified-smotebagging for pre-mirna classification," *Nucleic acids research*, vol. 41, no. 1, pp. e21–e21, 2013.

[15] P. Yang, P. D. Yoo, J. Fernando, B. B. Zhou, Z. Zhang, and A. Y. Zomaya, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE transactions on cybernetics*, vol. 44, no. 3, pp. 445–455, 2013.

[16] P. Cao, J. Yang, W. Li, D. Zhao, and O. Zaiane, "Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule cad," *Computerized Medical Imaging and Graphics*, vol. 38, no. 3, pp. 137–150, 2014.

[17] U. R. Acharya, P. Chowriappa, H. Fujita, S. Bhat, S. Dua, J. E. Koh, L. Eugene, P. Kongmebhol, and K. H. Ng, "Thyroid lesion classification in 242 patient population using gabor transform features from high resolution ultrasound images," *Knowledge-Based Systems*, vol. 107, pp. 235–245, 2016.

[18] B. Krawczyk, G. Schaefer, and M. Woźniak, "A hybrid cost-sensitive ensemble for imbalanced breast thermogram classification," *Artificial intelligence in medicine*, vol. 65, no. 3, pp. 219–227, 2015.

[19] W. Lu, Z. Li, and J. Chu, "A novel computer-aided diagnosis system for breast mri based on feature selection and ensemble learning," *Computers in biology and medicine*, vol. 83, pp. 157–165, 2017.

[20] J. A. Sanz, D. Bernardo, F. Herrera, H. Bustince, and H. Hagras, "A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 973–990, 2014.

[21] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

[22] B. Zhu, B. Baesens, and S. K. vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information sciences*, vol. 408, pp. 84–99, 2017.

[23] Q. D. Tran and P. Liatsis, "Raboc: An approach to handle class imbalance in multimodal biometric authentication," *Neurocomputing*, vol. 188, pp. 167–177, 2016.

[24] P. V. Radtke, E. Granger, R. Sabourin, and D. O. Gorodnichy, "Skew-sensitive boolean combination for adaptive ensembles–an application to face recognition in video surveillance," *Information Fusion*, vol. 20, pp. 31–48, 2014.

[25] R. Soleymani, E. Granger, and G. Fumera, "Loss factors for learning boosting ensembles from imbalanced data," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 204–209.

[26] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura, "Early dropout prediction using data mining: a case study with high school students," *Expert Systems*, vol. 33, no. 1, pp. 107–124, 2016.

[27] K. Ting, S. Tan, and F. Liu, "Mass: A new ranking measure for anomaly detection," *Gippsland School of Information Technology, Monash University*, 2009.

[28] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[29] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[30] C. L. Krumhansl, "Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density." *American Psychologist*, vol. 5, pp. 445–463, 1978.

[31] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. Singapore: Springer, 2016, pp. 1205–1214.

[32] A. Hoang, T. N. Mau, D. V. Vo, and V. N. Huynh, "A mass-based approach for local outlier detection," *IEEE Access*, vol. 9, pp. 16 448–16 466, 2021.

[33] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," in *Classic works of the Dempster-Shafer theory of belief functions*. Springer, 2008, pp. 57–72.

[34] G. Shafer, *A mathematical theory of evidence*. Princeton university press, 1976, vol. 42.

[35] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. New York: Springer, 1992, pp. 196–202.

[36] D. Hawkins, "Bayesian approach to outliers," in *Identification of Outliers*. Springer, 1980, pp. 115–122.

[37] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020.

[38] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Networks*, vol. 90, p. 101842, 2019.

[39] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, 2019.

[40] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *2018 IEEE 15th International*

Conference on Networking, Sensing and Control (ICNSC). IEEE, 2018, pp. 1–6.

[41] A. D. Pawar, P. N. Kalavadekar, and S. N. Tambe, "A survey on outlier detection techniques for credit card fraud detection," *IOSR Journal of Computer Engineering*, vol. 16, no. 2, pp. 44–48, 2014.

[42] X. Zhang, D. Ma, H. Yu, Y. Huang, P. Howell, and B. Stevens, "Scene perception guided crowd anomaly detection," *Neurocomputing*, vol. 414, pp. 291–302, 2020.

[43] L. Liu, G. Han, Y. He, and J. Jiang, "Fault-tolerant event region detection on trajectory pattern extraction for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2072–2080, 2019.

[44] C.-H. Lin, K.-C. Hsu, K. R. Johnson, M. Luby, and Y. C. Fann, "Applying density-based outlier identifications using multiple datasets for validation of stroke clinical outcomes," *International journal of medical informatics*, vol. 132, p. 103988, 2019.

[45] A. Borah and B. Nath, "Incremental rare pattern based approach for identifying outliers in medical data," *Applied Soft Computing*, vol. 85, p. 105824, 2019.

[46] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: Local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1649–1652.

[47] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289–307, 2019.

[48] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais *et al.*, "Deep learning and process understanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, 2019.

[49] W. D. Fisher, T. K. Camp, and V. V. Krzhizhanovskaya, "Anomaly detection in earth dam and levee passive seismic data using support vector machines and automatic feature selection," *Journal of Computational Science*, vol. 20, pp. 143–153, 2017.

[50] A. Cano, A. Zafra, and S. Ventura, "Weighted data gravitation classification for standard and imbalanced data," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1672–1687, 2013.

[51] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, T. Washio, and Z.-H. Zhou, "Lowest probability mass neighbour algorithms: Relaxing the metric constraint in distance-based neighbourhood algorithms," *Machine Learning*, vol. 108, no. 2, pp. 331–376, 2019.

[52] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[53] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[54] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[55] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[56] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107 964–108 000, 2019.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[58] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[59] I. Triguero, S. González, J. M. Moyano, S. García, J. Alcalá-Fdez, J. Luengo, A. Fernández, M. J. del Jesús, L. Sánchez, and F. Herrera, "Keel 3.0: an open source software for multi-stage analysis in data mining," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 1238–1249, 2017.

[60] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[61] F. Keller, E. Muller, and K. Bohm, "Hics: High contrast subspaces for density-based outlier ranking," in *2012 IEEE 28th international conference on data engineering*. IEEE, 2012, pp. 1037–1048.

[62] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *Acm Sigkdd Explorations Newsletter*, vol. 17, no. 1, pp. 24–47, 2015.

[63] B. Micenková, B. McWilliams, and I. Assent, "Learning outlier ensembles: The best of both worlds–supervised and unsupervised," in *Proceedings of the ACM SIGKDD 2014 Workshop on Outlier Detection and Description under Data Diversity (ODD2). New York, NY, USA*. Citeseer, 2014, pp. 51–54.

[64] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *Acm transactions on knowledge discovery from data (tkdd)*, vol. 10, no. 4, pp. 1–33, 2016.

[65] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 2015, pp. 159–166.

[66] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert systems with applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[67] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.

[68] B. Lebichot, Y.-A. Le Borgne, L. He-Guelton, F. Oblé, and G. Bontempi, "Deep-learning domain adaptation techniques for credit cards fraud detection," in *INNS Big Data and Deep Learning conference*. Springer, 2019, pp. 78–88.

[69] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[70] D. A. Cieslak and N. V. Chawla, "Learning decision trees for unbalanced data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Antwerp: Springer, 2008, pp. 241–256.

[71] J.-S. Lee, "Auc4. 5: Auc-based c4. 5 decision tree algorithm for imbalanced data classification," *IEEE Access*, vol. 7, pp. 106 034–106 042, 2019.

[72] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, vol. 18, no. 60, pp. 1–8, 2006.

[73] C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis, "Uncertainty based under-sampling for learning naive bayes classifiers under imbalanced data sets," *IEEE Access*, vol. 8, pp. 2122–2133, 2019.

[74] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Berlin: Springer, 2003, pp. 986–996.

[75] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, pp. 1–19, 2017.

[76] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[77] A. De Caigny, K. Coussement, and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees," *European Journal of Operational Research*, vol. 269, no. 2, pp. 760–772, 2018.

[78] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[79] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[80] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in

Proceedings of the 25th international conference on Machine learning. Helsinki, Finland: Springer, 2008, pp. 408–415.

[81] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear svm: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803–855, 2019.

[82] M. Ring and B. M. Eskofier, "An approximation of the gaussian rbf kernel for efficient classification with svms," *Pattern Recognition Letters*, vol. 84, pp. 107–113, 2016.

[83] S. E. Roshan and S. Asadi, "Improvement of bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103319, 2020.

[84] L. Guo, S. Boukir, and A. Aussem, "Building bagging on critical instances," *Expert Systems*, vol. 37, no. 2, p. e12486, 2020.

[85] J. Hatwell, M. M. Gaber, and R. M. A. Azad, "Ada-whips: explaining adaboost classification with applications in the health sciences," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–25, 2020.

[86] K. M. Asim, A. Idris, T. Iqbal, and F. Martínez-Álvarez, "Seismic indicators based earthquake predictor system using genetic programming and adaboost classification," *Soil Dynamics and Earthquake Engineering*, vol. 111, pp. 1–7, 2018.

[87] X. Ren, H. Guo, S. Li, S. Wang, and J. Li, "A novel image classification method with cnn-xgboost model," in *International Workshop on Digital Watermarking*. Magdeburg, Saxony-Anhalt: Springer, 2017, pp. 378–390.

[88] C. Wang, C. Deng, and S. Wang, "Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost," *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020.

[89] M. E. Kadir, P. S. Akash, S. Sharmin, A. A. Ali, and M. Shoyaib, "A proximity weighted evidential k nearest neighbor classifier for imbalanced data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Delhi, India: Springer, 2020, pp. 71–83.

[90] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[91] D. Devi, B. Purkayastha *et al.*, "Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance," *Pattern Recognition Letters*, vol. 93, pp. 3–12, 2017.

[92] M. Lichman *et al.*, "Uci machine learning repository," 2013.

[93] J. P. Siebert, "Vehicle recognition using rule based methods," *Turing Institute Research Memorandum TIRM-87-0.18*, 1987.

[94] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Bioinformatics*, vol. 11, no. 2, pp. 95–110, 1991.

Appendix A. Plots of ROC curves and PR curves on the Glass, Cardio, Shuttle, and Parkinson datasets for comparison purposes in Chapter 3.

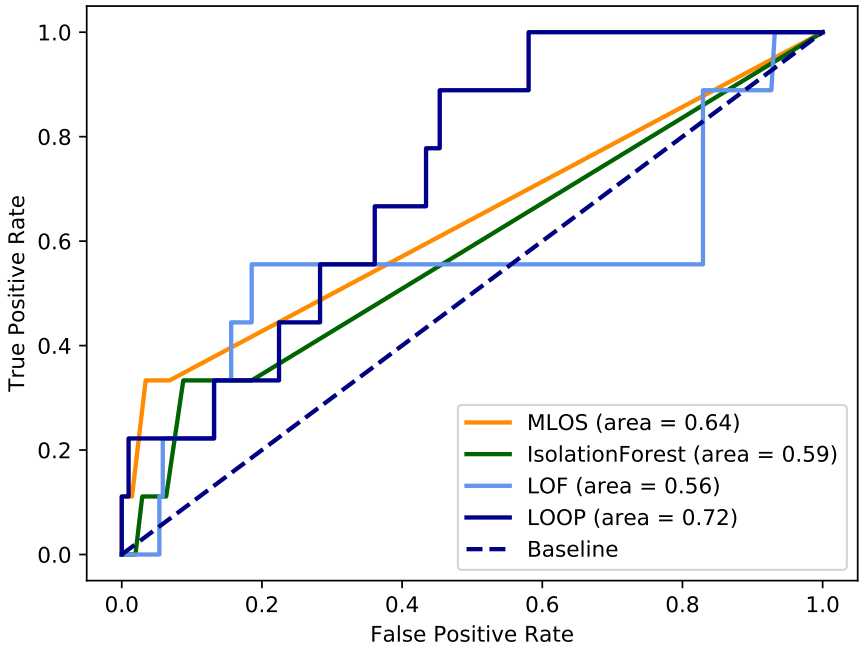


Figure 5.1: ROC curves tested on the Glass dataset.

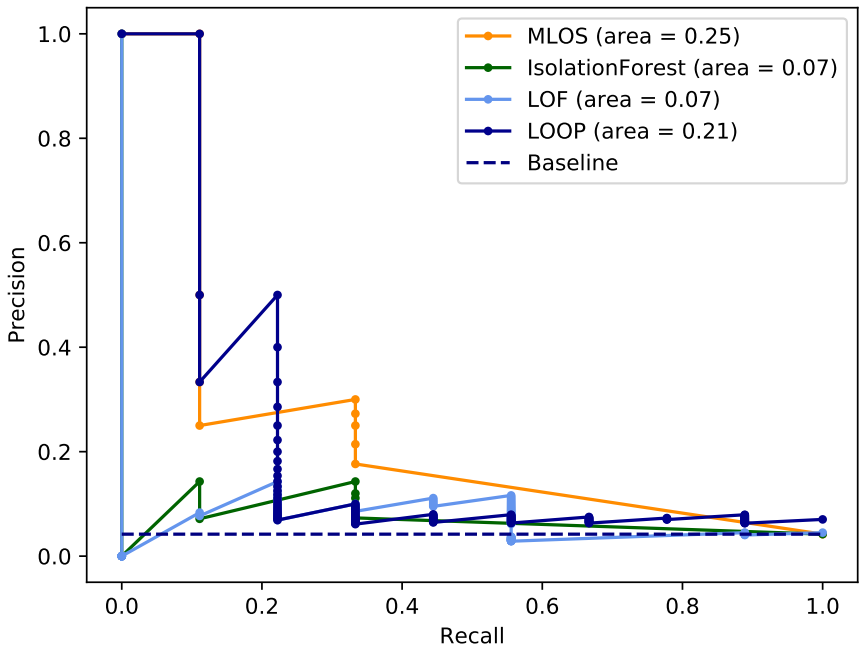


Figure 5.2: PR curves tested on the Glass dataset.

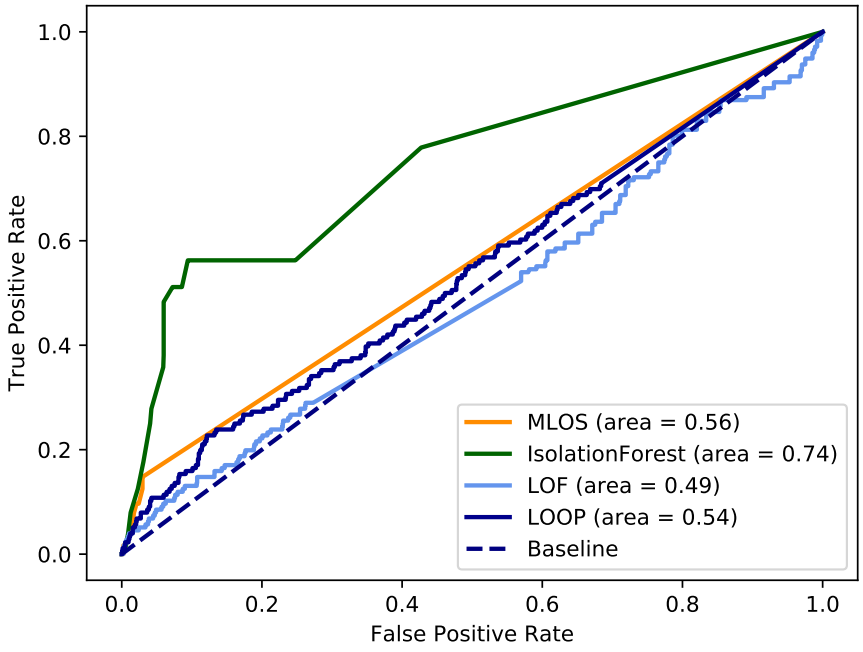


Figure 5.3: ROC curves tested on the Cardio dataset.

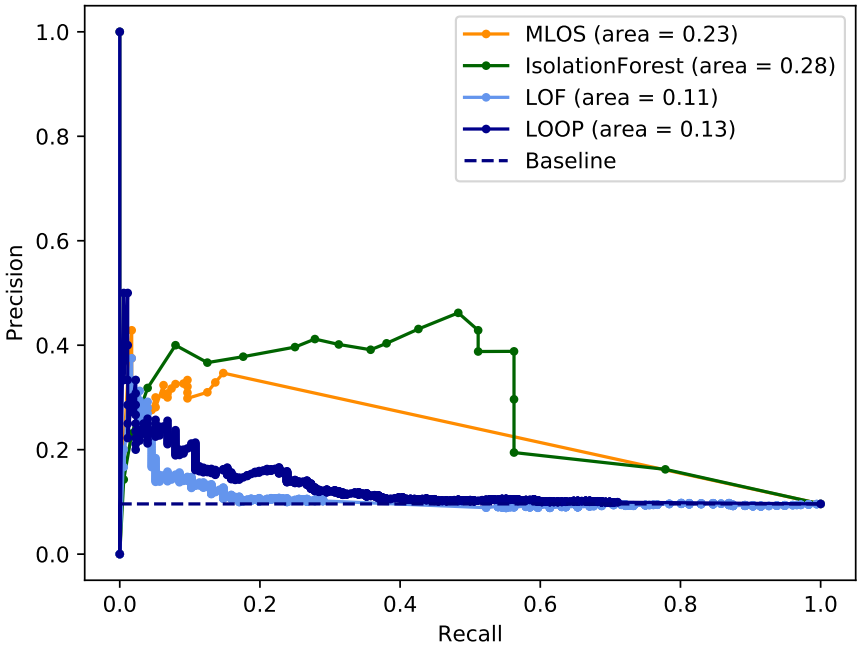


Figure 5.4: PR curves tested on the Cardio dataset.

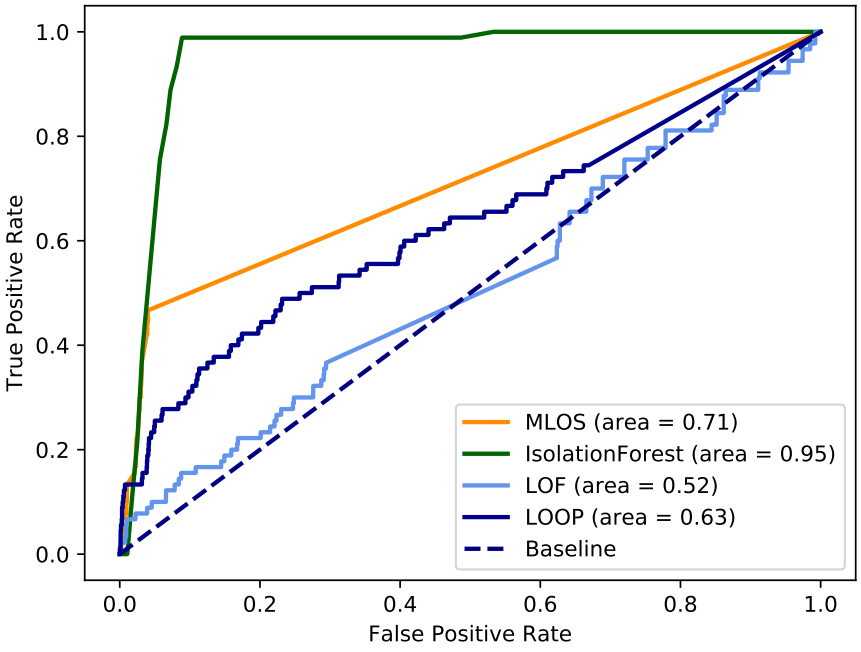


Figure 5.5: ROC curves tested on the Shuttle dataset.

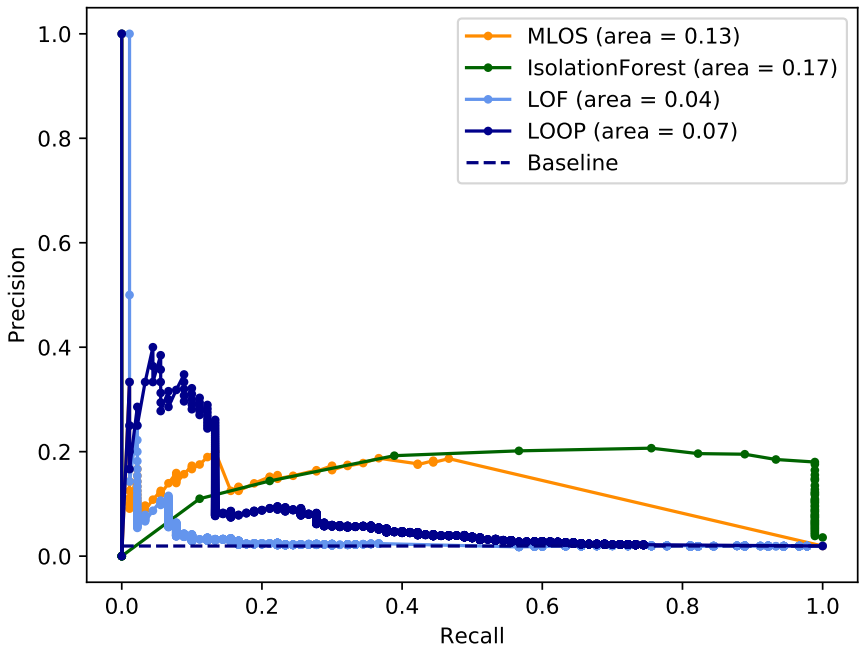


Figure 5.6: PR curves tested on the Shuttle dataset.

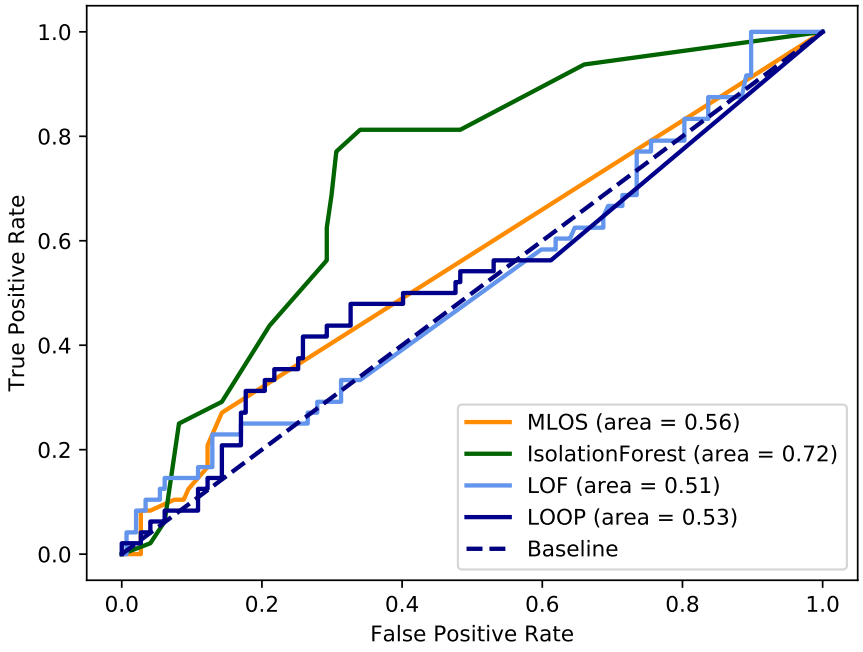


Figure 5.7: ROC curves tested on the Parkinsons dataset.

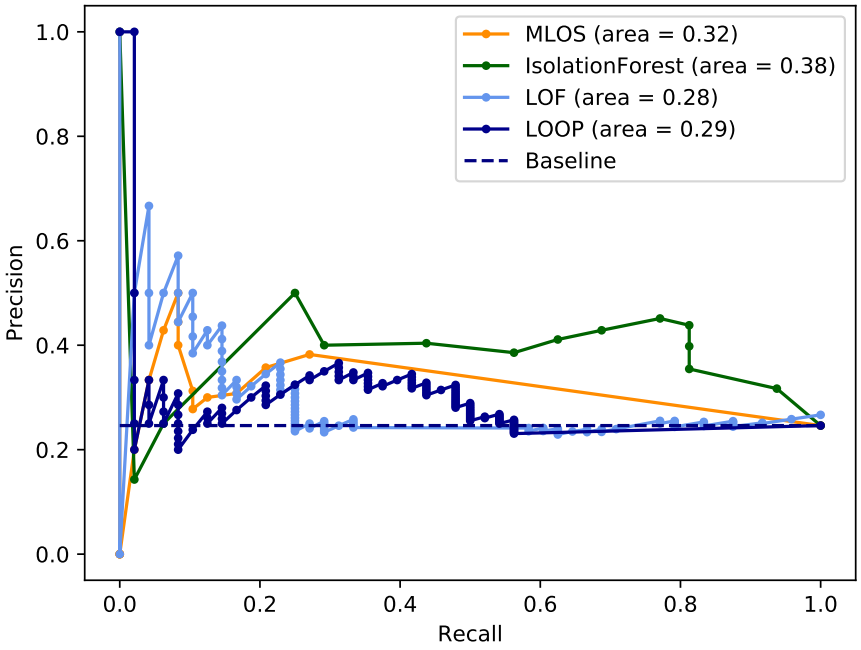


Figure 5.8: PR curves tested on the Parkinsons dataset.

Appendix B. Detailed comparison results in Chapter ??.

Table 5.1: F1 score results comparisons.

Idx.	Dataset	DT	NB	kNN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPEkNN	EMass
1	Glass1	0.722	0.591	0.581	0.526	0.692	0.591	0.491	0.688	0.667	0.647	0.650	0.778
2	Wisconsin	0.952	0.962	0.961	0.990	0.944	0.981	0.971	0.953	0.981	0.952	0.981	0.981
3	Pima	0.639	0.684	0.565	0.624	0.656	0.624	0.677	0.703	0.651	0.646	0.639	0.614
4	Ecoli-0_vs_1	0.963	0.783	1.000	1.000	0.963	1.000	1.000	0.963	0.963	0.963	0.963	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.667	0.556	0.545	0.483	0.692	0.516	0.407	0.692	0.741	0.769	0.581	0.688
7	Vehicle2	0.819	0.583	0.833	0.927	0.819	0.911	0.509	0.819	0.953	0.952	0.800	0.962
8	Vehicle1	0.615	0.523	0.483	0.693	0.504	0.710	0.513	0.667	0.585	0.673	0.558	0.647
9	Glass-0-1-2-3_vs_4-5-6	0.846	0.786	0.889	0.897	0.769	0.889	0.517	0.929	1.000	0.889	0.929	1.000
10	Vehicle0	0.822	0.609	0.882	0.946	0.643	0.946	0.622	0.643	0.939	0.909	0.909	0.878
11	Ecoli1	0.733	0.343	0.667	0.690	0.759	0.733	0.714	0.710	0.640	0.769	0.690	0.714
12	New-thyroid1	1.000	0.923	0.909	1.000	1.000	1.000	0.909	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.941	1.000	1.000	0.933	0.941	0.933	0.875	0.941	1.000	0.941	1.000	1.000
14	Ecoli2	0.783	0.339	0.818	0.690	0.741	0.667	0.783	0.692	0.783	0.818	0.750	0.720
15	Segment0	0.970	0.615	0.969	0.992	0.590	0.992	0.558	0.955	0.985	0.962	0.961	0.984
16	Glass6	0.750	0.875	0.857	0.762	0.778	0.889	0.000	0.857	0.857	0.875	0.800	0.857
17	Yeast3	0.850	0.272	0.800	0.694	0.698	0.687	0.764	0.791	0.778	0.840	0.708	0.735
18	Ecoli3	0.636	0.516	0.615	0.556	0.533	0.615	0.667	0.667	0.706	0.769	0.706	0.778
19	Page-blocks0	0.589	0.500	0.805	0.736	0.631	0.688	0.275	0.585	0.813	0.850	0.768	0.822
20	Yeast-0-2-5-7-9_vs_3-6-8	0.741	0.230	0.851	0.656	0.704	0.702	0.741	0.741	0.755	0.667	0.741	0.755
21	Yeast-0-2-5-6_vs_3-7-8-9	0.615	0.313	0.700	0.471	0.582	0.516	0.571	0.615	0.576	0.566	0.595	0.615
22	Yeast-2_vs_4	0.710	0.818	0.842	0.667	0.846	0.692	0.750	0.667	0.880	0.645	0.696	0.900
23	Ecoli-0-6-7_vs_3-5	0.857	0.000	1.000	0.600	0.600	0.667	0.667	1.000	0.800	0.857	0.857	0.800
24	Ecoli-0-1_vs_2-3-5	0.267	0.000	1.000	0.500	0.500	0.444	0.800	0.800	0.571	0.500	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.615	0.444	0.889	0.727	0.667	0.727	0.727	0.889	0.750	0.889	0.889	0.889
26	Ecoli-0-2-6-7_vs_3-5	0.889	0.000	1.000	0.533	0.667	0.667	0.615	0.889	0.800	0.727	0.857	0.857
27	Ecoli-0-4-6_vs_5	0.800	0.750	0.857	0.750	0.857	0.750	0.857	0.667	0.857	0.800	0.857	0.857
28	Ecoli-0-3-4-7_vs_5-6	0.588	0.400	0.909	0.625	0.714	0.769	0.833	0.800	1.000	0.625	0.667	0.769
29	Ecoli-0-3-4-6_vs_5	0.462	0.727	0.889	0.667	0.889	0.727	0.800	0.600	0.800	0.500	0.889	0.889
30	Vowel0	0.875	0.792	1.000	0.724	0.731	0.724	0.828	0.875	0.958	0.875	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	0.667	0.800	1.000	0.800	0.000	1.000	1.000	1.000	1.000	0.667
32	Ecoli-0-6-7_vs_5	0.615	1.000	0.857	0.500	0.800	0.800	0.889	0.889	1.000	1.000	0.800	0.667
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.600	0.444	0.667	0.533	0.533	0.533	0.625	0.667	0.571	0.667	0.500	0.600
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.769	0.667	0.800	0.625	0.647	0.769	0.769	0.769	0.625	0.710	0.783	0.783
35	Ecoli-0-1_vs_5	0.833	0.833	0.923	0.875	1.000	0.857	0.923	0.833	0.833	0.769	0.923	0.923
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.286	0.400	0.000	1.000	1.000	1.000	0.667	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.444	0.750	0.667	0.571	0.400	0.429	0.600	0.444	0.727	0.444	0.667	0.750
38	Ecoli-0-1-4-6_vs_5	0.600	0.889	1.000	0.571	0.889	0.571	0.889	0.857	0.800	0.667	0.889	1.000
39	Shuttle-c0-vs-c4	1.000	0.967	0.983	0.983	1.000	1.000	0.968	1.000	1.000	1.000	0.983	1.000
40	Page-blocks-1-3_vs_4	0.947	0.526	0.800	0.857	0.581	0.783	0.455	0.947	0.947	0.947	0.778	0.941
41	Glass4	0.400	0.000	1.000	0.667	0.667	0.800	0.170	0.750	0.667	0.750	0.800	1.000
42	Dermatology-6	1.000	0.333	1.000	1.000	1.000	1.000	0.444	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.500	0.000	0.000	0.667	0.000	0.667	0.250	0.667	0.000	0.400	0.000	0.667
44	Ecoli4	1.000	0.522	1.000	0.857	0.667	0.800	0.800	1.000	0.833	1.000	0.923	1.000
45	Zoo-3	0.333	0.667	0.000	0.667	0.400	0.667	0.400	0.000	0.667	0.667	1.000	0.667
46	Poker-9_vs_7	0.222	0.667	0.667	0.000	0.333	0.000	0.800	0.667	0.667	0.500	0.500	0.667
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.074	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.800
49	Yeast-2_vs_8	0.169	0.800	0.800	0.800	0.667	0.533	0.615	0.727	0.667	0.727	0.667	0.800
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	0.889	0.889	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.593	0.168	0.875	0.615	0.485	0.471	0.500	0.593	0.632	0.750	0.700	0.609
53	Ecoli-0-1-3-7_vs_2-6	0.286	0.667	1.000	0.286	0.500	0.286	0.333	0.667	0.667	0.000	0.667	0.667
54	Abalone-21_vs_8	0.571	0.381	0.571	0.889	0.348	0.800	0.727	0.800	0.667	0.444	0.500	0.750
55	Kddcup-land_vs_portsweep	1.000	0.010	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.025	0.000	0.000	0.035	0.000	0.020	0.000	0.000	0.030	0.067	0.000	1.000
57	Shuttle-2_vs_5	1.000	0.116	1.000	1.000	1.000	1.000	0.667	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.941	1.000	0.941	0.842	1.000	0.842	1.000	1.000	1.000	0.933	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.667	1.000	1.000	0.667	0.400	1.000	1.000	1.000	0.667	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	0.800	1.000	0.857	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.744	0.605	0.818	0.747	0.716	0.738	0.622	0.801	0.813	0.790	0.796	**0.857**
	Avg. Ranks	5.000	3.383	6.350	4.483	4.100	4.567	4.133	5.600	6.333	5.850	5.583	**7.100**

Table 5.2: Brier score results comparisons.

Idx.	Dataset	DT	NB	k-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPEkNN	EMass
1	Glass1	0.174	0.354	0.171	0.239	0.194	0.201	0.220	0.166	0.229	0.164	0.264	0.172
2	Wisconsin	0.034	0.029	0.025	0.012	0.044	0.016	0.013	0.032	0.163	0.059	0.015	0.015
3	Pima	0.192	0.172	0.240	0.190	0.212	0.173	0.168	0.176	0.239	0.183	0.221	0.241
4	Ecoli-0_vs_1	0.011	0.080	0.013	0.011	0.073	0.007	0.004	0.009	0.100	0.049	0.017	0.001
5	Iris0	0.000	0.000	0.000	0.033	0.001	0.000	0.000	0.000	0.000	0.038	0.000	0.000
6	Glass0	0.172	0.354	0.147	0.186	0.155	0.151	0.193	0.124	0.198	0.122	0.263	0.227
7	Vehicle2	0.087	0.169	0.072	0.042	0.170	0.041	0.194	0.091	0.189	0.055	0.079	0.024
8	Vehicle1	0.155	0.279	0.192	0.126	0.221	0.131	0.185	0.153	0.236	0.140	0.208	0.177
9	Glass-0-1-2-3_vs_4-5-6	0.094	0.120	0.062	0.055	0.097	0.067	0.251	0.056	0.135	0.079	0.046	0.003
10	Vehicle0	0.083	0.299	0.042	0.015	0.165	0.026	0.132	0.123	0.188	0.070	0.050	0.059
11	Ecoli1	0.097	0.657	0.096	0.100	0.126	0.081	0.082	0.090	0.168	0.092	0.131	0.111
12	New-thyroid1	0.002	0.008	0.013	0.003	0.037	0.008	0.047	0.004	0.051	0.040	0.006	0.002
13	Newthyroid2	0.023	0.000	0.005	0.023	0.025	0.012	0.043	0.016	0.030	0.050	0.006	0.000
14	Ecoli2	0.073	0.560	0.057	0.108	0.142	0.087	0.054	0.073	0.177	0.073	0.092	0.103
15	Segment0	0.008	0.161	0.009	0.002	0.115	0.004	0.042	0.013	0.084	0.040	0.010	0.004
16	Glass6	0.100	0.044	0.054	0.073	0.061	0.102	0.153	0.035	0.136	0.055	0.065	0.033
17	Yeast3	0.056	0.610	0.045	0.075	0.144	0.052	0.043	0.055	0.177	0.070	0.084	0.079
18	Ecoli3	0.079	0.208	0.056	0.081	0.127	0.067	0.055	0.070	0.162	0.062	0.072	0.062
19	Page-blocks0	0.075	0.090	0.032	0.067	0.123	0.072	0.074	0.091	0.210	0.056	0.041	0.034
20	Yeast-0-2-5-6_vs_3-7-8-9	0.137	0.091	0.059	0.156	0.183	0.067	0.059	0.118	0.234	0.129	0.069	0.071
21	Yeast-0-2-5-7-9_vs_3-6-8	0.076	0.532	0.033	0.099	0.144	0.038	0.030	0.067	0.216	0.094	0.056	0.055
22	Yeast-2_vs_4	0.062	0.034	0.032	0.092	0.110	0.029	0.027	0.052	0.189	0.076	0.046	0.024
23	Ecoli-0-6-7_vs_3-5	0.052	0.088	0.010	0.121	0.111	0.044	0.013	0.020	0.189	0.059	0.006	0.021
24	Ecoli-0-1_vs_2-3-5	0.104	0.040	0.002	0.070	0.110	0.025	0.005	0.037	0.181	0.071	0.000	0.000
25	Ecoli-0-2-3-4_vs_5	0.104	0.204	0.011	0.082	0.081	0.066	0.029	0.029	0.110	0.064	0.024	0.024
26	Ecoli-0-2-6-7_vs_3-5	0.064	0.108	0.005	0.093	0.119	0.047	0.012	0.027	0.186	0.068	0.020	0.024
27	Ecoli-0-4-6_vs_5	0.024	0.037	0.014	0.070	0.059	0.059	0.020	0.041	0.144	0.062	0.021	0.024
28	Ecoli-0-3-4-7_vs_5-6	0.079	0.240	0.015	0.088	0.117	0.031	0.013	0.036	0.164	0.084	0.022	0.042
29	Ecoli-0-3-4-6_vs_5	0.136	0.054	0.027	0.103	0.104	0.075	0.019	0.052	0.144	0.113	0.027	0.025
30	Vowel0	0.032	0.047	0.000	0.058	0.104	0.038	0.021	0.031	0.129	0.051	0.000	0.000
31	Glass-0-4_vs_5	0.000	0.000	0.023	0.053	0.017	0.008	0.098	0.000	0.000	0.039	0.012	0.026
32	Ecoli-0-6-7_vs_5	0.043	0.017	0.023	0.121	0.083	0.073	0.013	0.025	0.172	0.046	0.030	0.032
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.075	0.074	0.049	0.098	0.119	0.049	0.041	0.055	0.187	0.084	0.057	0.061
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.084	0.081	0.057	0.076	0.122	0.061	0.048	0.067	0.214	0.095	0.048	0.054
35	Ecoli-0-1_vs_5	0.042	0.029	0.021	0.035	0.050	0.093	0.022	0.035	0.087	0.074	0.021	0.021
36	Glass-0-6_vs_5	0.000	0.000	0.005	0.004	0.123	0.027	0.065	0.000	0.000	0.044	0.012	0.000
37	Ecoli-0-1-4-7_vs_5-6	0.073	0.034	0.038	0.069	0.100	0.047	0.039	0.053	0.111	0.078	0.039	0.023
38	Ecoli-0-1-4-6_vs_5	0.056	0.020	0.002	0.097	0.065	0.047	0.005	0.032	0.142	0.066	0.005	0.002
39	Shuttle-c0-vs-c4	0.000	0.005	0.003	0.006	0.009	0.000	0.003	0.000	0.000	0.034	0.003	0.000
40	Page-blocks-1-3_vs_4	0.011	0.086	0.026	0.030	0.095	0.044	0.060	0.009	0.011	0.042	0.026	0.007
41	Glass4	0.070	0.195	0.013	0.086	0.119	0.081	0.088	0.039	0.076	0.065	0.028	0.009
42	Dermatology-6	0.000	0.111	0.003	0.001	0.047	0.004	0.015	0.001	0.000	0.035	0.007	0.000
43	Winequality-white-9_vs_4	0.058	0.058	0.059	0.029	0.064	0.052	0.055	0.042	0.043	0.082	0.059	0.018
44	Ecoli4	0.000	0.155	0.005	0.015	0.086	0.005	0.005	0.006	0.096	0.037	0.015	0.000
45	Zoo-3	0.129	0.048	0.042	0.048	0.115	0.073	0.097	0.071	0.097	0.069	0.022	0.048
46	Poker-9_vs_7	0.116	0.026	0.016	0.140	0.118	0.039	0.024	0.050	0.021	0.071	0.020	0.012
47	Shuttle-c2-vs-c4	0.000	0.000	0.000	0.000	0.004	0.000	0.037	0.000	0.000	0.037	0.000	0.000
48	Shuttle-6_vs_2-3	0.000	0.000	0.000	0.031	0.038	0.003	0.001	0.000	0.000	0.036	0.000	0.009
49	Yeast-2_vs_8	0.178	0.021	0.027	0.246	0.169	0.028	0.026	0.092	0.195	0.076	0.029	0.025
50	Kddcup-guess_passwd_vs_satan	0.000	0.000	0.000	0.000	0.003	0.001	0.000	0.000	0.000	0.034	0.000	0.000
51	Abalone-3_vs_11	0.000	0.000	0.000	0.000	0.000	0.003	0.001	0.000	0.000	0.035	0.000	0.000
52	Yeast5	0.020	0.256	0.010	0.028	0.069	0.012	0.011	0.023	0.092	0.042	0.020	0.029
53	Ecoli-0-1-3-7_vs_2-6	0.053	0.018	0.004	0.075	0.064	0.016	0.014	0.026	0.069	0.049	0.010	0.018
54	Abalone-21_vs_8	0.037	0.093	0.028	0.009	0.096	0.018	0.021	0.040	0.156	0.057	0.031	0.019
55	Kddcup-land_vs_portsweep	0.000	0.906	0.000	0.000	0.003	0.001	0.005	0.000	0.000	0.034	0.000	0.000
56	Poker-8-9_vs_6	0.197	0.017	0.018	0.254	0.210	0.016	0.017	0.145	0.235	0.094	0.019	0.000
57	Shuttle-2_vs_5	0.000	0.070	0.001	0.000	0.005	0.000	0.005	0.000	0.000	0.034	0.001	0.000
58	Kddcup-buffer_overflow_vs_back	0.000	0.002	0.000	0.002	0.019	0.002	0.001	0.000	0.000	0.034	0.001	0.000
59	Kddcup-land_vs_satan	0.000	0.000	0.003	0.001	0.007	0.003	0.003	0.000	0.000	0.034	0.003	0.000
60	Kddcup-rootkit-imap_vs_back	0.000	0.000	0.000	0.001	0.026	0.001	0.000	0.000	0.000	0.034	0.000	0.000
	Avg. Values	0.060	0.133	**0.034**	0.067	0.092	0.044	0.050	0.045	0.113	0.066	0.041	**0.034**
	Avg. Ranks	6.633	5.083	8.817	5.483	3.150	7.233	7.233	7.750	3.717	4.650	7.667	**9.067**

Table 5.3: ROC-AUC results comparison.

Idx.	Dataset	DT	NB	k-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPEkNN	EMass
1	Glass1	0.841	0.677	0.813	0.626	0.800	0.709	0.330	0.855	0.791	0.877	0.892	0.961
2	Wisconsin	0.982	0.994	0.987	0.993	0.989	0.993	0.988	0.991	0.997	0.991	0.987	0.987
3	Pima	0.804	0.825	0.701	0.815	0.812	0.799	0.806	0.830	0.813	0.808	0.744	0.722
4	Ecoli-0_vs_1	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.825	0.781	0.832	0.807	0.896	0.815	0.287	0.925	0.940	0.912	0.841	0.895
7	Vehicle2	0.938	0.809	0.936	0.986	0.972	0.986	0.744	0.958	0.995	0.997	0.934	0.978
8	Vehicle1	0.815	0.679	0.740	0.892	0.663	0.891	0.690	0.856	0.836	0.875	0.754	0.823
9	Glass-0-1-2-3_vs_4-5-6	0.867	0.955	0.960	0.981	0.969	0.988	0.024	0.988	1.000	0.929	0.964	1.000
10	Vehicle0	0.947	0.807	0.984	0.998	0.839	0.995	0.857	0.943	0.998	0.993	0.980	0.979
11	Ecoli1	0.883	0.847	0.845	0.934	0.884	0.940	0.945	0.892	0.891	0.890	0.858	0.895
12	New-thyroid1	1.000	1.000	1.000	1.000	1.000	1.000	0.982	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.986	1.000	1.000	1.000	1.000	1.000	0.896	1.000	1.000	1.000	1.000	1.000
14	Ecoli2	0.860	0.861	0.892	0.890	0.915	0.874	0.895	0.902	0.957	0.909	0.889	0.893
15	Segment0	0.990	0.987	0.983	1.000	0.987	1.000	0.971	0.986	0.998	0.990	0.983	0.985
16	Glass6	0.850	0.971	0.864	0.982	0.989	0.986	0.050	0.982	0.975	0.993	0.864	0.986
17	Yeast3	0.947	0.966	0.938	0.968	0.980	0.962	0.977	0.973	0.967	0.973	0.935	0.972
18	Ecoli3	0.916	0.950	0.858	0.927	0.965	0.935	0.979	0.949	0.950	0.975	0.867	0.918
19	Page-blocks0	0.954	0.931	0.940	0.973	0.938	0.960	0.880	0.955	0.988	0.985	0.944	0.942
20	Yeast-0-2-5-6_vs_3-7-8-9	0.843	0.851	0.869	0.853	0.899	0.854	0.832	0.867	0.843	0.879	0.862	0.842
21	Yeast-0-2-5-7-9_vs_3-6-8	0.959	0.954	0.985	0.960	0.950	0.961	0.969	0.973	0.960	0.959	0.981	0.983
22	Yeast-2_vs_4	0.981	0.980	0.900	0.963	0.993	0.975	0.988	0.994	0.998	0.986	0.897	0.904
23	Ecoli-0-6-7_vs_3-5	0.988	0.976	1.000	0.976	0.984	0.976	1.000	1.000	1.000	0.992	1.000	1.000
24	Ecoli-0-1_vs_2-3-5	0.888	1.000	1.000	0.979	1.000	0.989	1.000	1.000	1.000	0.936	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.973	1.000	1.000	0.953	0.986	0.959	1.000	1.000	0.993	1.000	1.000	1.000
26	Ecoli-0-2-6-7_vs_3-5	1.000	0.951	1.000	0.976	1.000	0.970	1.000	1.000	0.976	0.994	1.000	1.000
27	Ecoli-0-4-6_vs_5	0.833	0.991	1.000	0.982	0.991	0.982	1.000	0.982	1.000	0.939	1.000	1.000
28	Ecoli-0-3-4-7_vs_5-6	0.964	0.966	0.998	0.983	0.996	1.000	1.000	0.991	1.000	0.991	0.987	0.985
29	Ecoli-0-3-4-6_vs_5	0.845	0.986	0.986	0.959	1.000	0.986	1.000	0.980	0.993	0.885	0.986	0.983
30	Vowel0	0.932	0.976	1.000	0.979	0.958	0.977	0.993	0.978	0.996	0.972	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
32	Ecoli-0-6-7_vs_5	1.000	1.000	0.991	0.981	1.000	1.000	1.000	1.000	1.000	1.000	0.994	0.981
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.761	0.941	0.742	0.828	0.938	0.812	0.884	0.935	0.952	0.886	0.742	0.817
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.918	0.962	0.896	0.954	0.964	0.934	0.924	0.960	0.957	0.958	0.899	0.936
35	Ecoli-0-1_vs_5	0.857	0.882	0.929	0.993	1.000	0.895	0.993	1.000	0.997	0.836	0.929	0.929
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.810	1.000	0.333	1.000	1.000	1.000	1.000	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.684	0.958	0.794	0.832	0.932	0.803	0.806	0.910	0.968	0.869	0.794	0.897
38	Ecoli-0-1-4-6_vs_5	0.868	0.981	1.000	0.962	0.995	0.962	1.000	0.971	1.000	0.844	1.000	1.000
39	Shuttle-c0-vs-c4	1.000	0.965	0.983	0.967	1.000	1.000	1.000	1.000	1.000	1.000	0.983	1.000
40	Page-blocks-1-3_vs_4	0.994	0.934	0.939	0.996	0.924	0.995	0.935	1.000	1.000	0.994	0.939	1.000
41	Glass4	0.625	0.750	1.000	0.936	0.974	0.949	0.071	0.981	0.801	0.955	1.000	1.000
42	Dermatology-6	1.000	0.979	1.000	1.000	1.000	1.000	0.993	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.734	0.891	0.500	1.000	0.984	0.969	0.844	0.938	0.969	0.891	0.500	1.000
44	Ecoli4	1.000	0.984	1.000	1.000	0.989	1.000	1.000	1.000	0.995	1.000	0.995	1.000
45	Zoo-3	0.671	0.750	1.000	0.658	0.763	1.000	0.132	0.921	0.697	1.000	1.000	0.750
46	Poker-9_vs_7	0.686	0.543	0.989	0.564	0.894	0.479	1.000	1.000	0.500	0.830	0.989	1.000
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
49	Yeast-2_vs_8	0.941	0.902	0.888	0.897	0.911	0.865	0.872	0.841	0.967	0.972	0.888	0.888
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.992	0.993	0.934	0.992	0.993	0.993	0.993	0.994	0.990	0.994	0.931	0.930
53	Ecoli-0-1-3-7_vs_2-6	1.000	0.991	1.000	0.982	0.982	0.946	0.964	1.000	1.000	0.938	1.000	0.991
54	Abalone-21_vs_8	0.896	0.895	0.695	0.975	0.888	0.941	0.939	0.849	0.793	0.857	0.695	0.798
55	Kddcup-land_vs_portsweep	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.500	0.325	0.495	0.504	0.406	0.690	0.253	0.432	0.521	0.531	0.495	1.000
57	Shuttle-2_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.996	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.999	1.000	1.000	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.998	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.912	0.922	0.930	0.939	0.945	0.945	0.800	0.958	0.949	0.950	0.932	**0.959**
	Avg. Ranks	3.567	3.967	4.133	4.933	5.467	4.983	4.317	5.967	**6.283**	5.600	4.117	5.300

Table 5.4: PR-AUC results comparison.

Idx.	Dataset	DT	NB	k-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPEkNN	EMass
1	Glass1	0.795	0.472	0.779	0.372	0.632	0.432	0.242	0.656	0.537	0.807	0.850	0.921
2	Wisconsin	0.960	0.988	0.976	0.982	0.967	0.982	0.921	0.979	0.994	0.981	0.976	0.977
3	Pima	0.719	0.711	0.601	0.711	0.713	0.655	0.664	0.695	0.675	0.697	0.632	0.614
4	Ecoli-0_vs_1	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.501	0.537	0.698	0.585	0.784	0.599	0.178	0.759	0.820	0.775	0.715	0.795
7	Vehicle2	0.879	0.708	0.917	0.964	0.904	0.966	0.659	0.914	0.988	0.995	0.910	0.980
8	Vehicle1	0.361	0.547	0.536	0.728	0.545	0.730	0.557	0.667	0.729	0.698	0.563	0.666
9	Glass-0-1-2-3_vs_4-5-6	0.913	0.898	0.966	0.958	0.937	0.979	0.201	0.980	1.000	0.948	0.974	1.000
10	Vehicle0	0.902	0.555	0.955	0.995	0.719	0.985	0.696	0.879	0.994	0.981	0.941	0.940
11	Ecoli1	0.777	0.573	0.710	0.811	0.676	0.832	0.753	0.783	0.731	0.785	0.761	0.816
12	New-thyroid1	1.000	1.000	1.000	1.000	1.000	1.000	0.930	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.944	1.000	1.000	1.000	1.000	1.000	0.901	1.000	1.000	1.000	1.000	1.000
14	Ecoli2	0.785	0.747	0.828	0.746	0.828	0.601	0.822	0.818	0.856	0.877	0.816	0.841
15	Segment0	0.983	0.954	0.977	1.000	0.954	1.000	0.896	0.971	0.994	0.988	0.977	0.987
16	Glass6	0.795	0.808	0.857	0.915	0.964	0.926	0.101	0.940	0.940	0.969	0.857	0.944
17	Yeast3	0.852	0.863	0.811	0.752	0.856	0.725	0.773	0.854	0.754	0.850	0.788	0.882
18	Ecoli3	0.851	0.727	0.756	0.714	0.772	0.675	0.881	0.803	0.741	0.911	0.829	0.807
19	Page-blocks0	0.818	0.621	0.860	0.814	0.782	0.692	0.617	0.772	0.901	0.891	0.864	0.887
20	Yeast-0-2-5-6_vs_3-7-8-9	0.442	0.418	0.676	0.459	0.761	0.474	0.584	0.671	0.413	0.686	0.603	0.606
21	Yeast-0-2-5-7-9_vs_3-6-8	0.603	0.656	0.877	0.632	0.889	0.654	0.911	0.863	0.760	0.810	0.851	0.872
22	Yeast-2_vs_4	0.883	0.869	0.864	0.894	0.947	0.886	0.919	0.959	0.983	0.919	0.834	0.886
23	Ecoli-0-6-7_vs_3-5	0.875	0.514	1.000	0.514	0.850	0.514	1.000	1.000	1.000	0.903	1.000	1.000
24	Ecoli-0-1_vs_2-3-5	0.438	1.000	1.000	0.708	1.000	0.792	1.000	1.000	1.000	0.597	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.833	1.000	1.000	0.626	0.908	0.650	1.000	1.000	0.944	1.000	1.000	1.000
26	Ecoli-0-2-6-7_vs_3-5	1.000	0.455	1.000	0.579	1.000	0.544	1.000	1.000	0.796	0.944	1.000	1.000
27	Ecoli-0-4-6_vs_5	0.846	0.903	1.000	0.850	0.903	0.850	1.000	0.850	1.000	0.754	1.000	1.000
28	Ecoli-0-3-4-7_vs_5-6	0.775	0.675	0.983	0.835	0.963	1.000	1.000	0.938	1.000	0.938	0.920	0.843
29	Ecoli-0-3-4-6_vs_5	0.710	0.908	0.900	0.519	1.000	0.908	1.000	0.884	0.944	0.793	0.900	0.850
30	Vowel0	0.910	0.872	1.000	0.852	0.849	0.845	0.919	0.873	0.984	0.948	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.054	1.000	1.000	1.000	1.000	1.000
32	Ecoli-0-6-7_vs_5	1.000	1.000	0.946	0.835	1.000	1.000	1.000	1.000	1.000	1.000	0.944	0.835
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.461	0.658	0.672	0.656	0.687	0.643	0.691	0.712	0.795	0.647	0.672	0.635
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.825	0.789	0.819	0.791	0.791	0.764	0.762	0.830	0.781	0.834	0.807	0.795
35	Ecoli-0-1_vs_5	0.878	0.880	0.939	0.966	1.000	0.883	0.966	1.000	0.981	0.780	0.939	0.939
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.100	1.000	0.033	1.000	1.000	1.000	1.000	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.472	0.759	0.735	0.771	0.605	0.668	0.682	0.561	0.821	0.523	0.735	0.874
38	Ecoli-0-1-4-6_vs_5	0.821	0.579	1.000	0.462	0.944	0.462	1.000	0.842	1.000	0.782	1.000	1.000
39	Shuttle-c0-vs-c4	1.000	0.953	0.985	0.969	1.000	1.000	0.999	1.000	1.000	1.000	0.985	1.000
40	Page-blocks-1-3_vs_4	0.950	0.739	0.904	0.966	0.467	0.958	0.687	1.000	1.000	0.950	0.896	1.000
41	Glass4	0.660	0.155	1.000	0.399	0.579	0.442	0.051	0.767	0.775	0.804	1.000	1.000
42	Dermatology-6	1.000	0.700	1.000	1.000	1.000	1.000	0.792	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.515	0.587	0.529	1.000	0.792	0.708	0.149	0.633	0.417	0.466	0.529	1.000
44	Ecoli4	1.000	0.663	1.000	1.000	0.930	1.000	1.000	1.000	0.955	1.000	0.944	1.000
45	Zoo-3	0.399	0.774	1.000	0.551	0.570	1.000	0.053	0.663	0.557	1.000	1.000	0.774
46	Poker-9_vs_7	0.332	0.517	0.875	0.052	0.149	0.037	1.000	1.000	0.515	0.542	0.792	1.000
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.019	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
49	Yeast-2_vs_8	0.809	0.739	0.762	0.845	0.717	0.769	0.711	0.625	0.521	0.778	0.762	0.680
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.803	0.733	0.648	0.779	0.887	0.777	0.759	0.841	0.635	0.852	0.626	0.733
53	Ecoli-0-1-3-7_vs_2-6	1.000	0.750	1.000	0.250	0.250	0.125	0.167	1.000	1.000	0.083	1.000	0.750
54	Abalone-21_vs_8	0.849	0.652	0.563	0.849	0.627	0.824	0.721	0.811	0.642	0.580	0.563	0.759
55	Kddcup-land_vs_portsweep	1.000	1.000	1.000	1.000	1.000	1.000	0.002	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.215	0.012	0.008	0.018	0.013	0.030	0.010	0.016	0.017	0.024	0.008	1.000
57	Shuttle-2_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.359	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.944	1.000	1.000	0.915	1.000	1.000	1.000	1.000	1.000	1.000	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.750	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.819	0.776	0.878	0.795	0.819	0.800	0.697	0.880	0.865	0.852	0.875	**0.915**
	Avg. Ranks	4.250	3.600	5.067	4.517	5.000	4.567	4.017	5.767	5.917	5.817	4.933	**6.183**