PhD-FSTM-2020-65
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 29/10/2020 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Georgios KAIAFAS
Born on 1 September 1984 in Athens (Greece)

ENSEMBLE LEARNING FOR ANOMALY DETECTION WITH APPLICATIONS FOR CYBERSECURITY AND TELECOMMUNICATION

Dissertation defence committee

Dr Radu State, dissertation supervisor
A-Professor, Université du Luxembourg

Dr Valtchev Petko
Professor, University of Montreal

Dr-Ing Holger Voos, Chairman
Professor, Université du Luxembourg

Dr Sofiane Lagraa
Research Associate, Université du Luxembourg

Dr Sotiris Kotsiantis, Vice Chairman
Professor, University of Patras

To my loving Dad
To my loving Natani

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Georgios KAIAFAS
November 2020

Acknowledgements

I would like to thank particularly Dr. Habil. Radu State for having given me the opportunity to pursue a PhD at the university of Luxembourg in collaboration with POST Luxembourg. I am grateful for the time and the dedication they put to ensure the successfulness of the research project. I would like also to thank Dr. Sofiane Lagraa for his precise advice and challenging questions during the PhD supervision. Furthermore, I would like to thank Dr. Arthur Zimek for having given me the opportunity to do a research stay within the Data Science and Statistics group at the University of Southern Denmark. Additionally, I would like to thank Dr. Sotirios Kotsiantis for having given me the opportunity to do a research stay within the Mathematics department at the university of Patras. Finally, I would like to express my deep gratitude to my partner, Natani, and my family for their unconditional support during this long journey. I would like to thank all my fellow colleagues of the university of Luxembourg, the University of Southern Denmark, university of Patras and POST Luxembourg as well as all the others I did not mention who helped to contribute to this work.

Abstract

Nowadays cyber and telecommunication criminal activities are becoming more sophisticated and hazardous. Often, adversaries form large teams composed of hundreds of highly skilled members to raise the level of sophistication and perform well-organized attacks. As such, enterprises face enormous difficulties to detect such attacks and this is confirmed by several studies. The lateral movement attack is a stealthy, persistent and well-organized attack that mainly targets organizations and institutions to exfiltrate sensitive and valuable data. In addition, in the telecommunication industry, it is a matter of major concern to enterprises PBX fraud activities that allow adversaries to make free calls and gain financial benefit.

In this thesis, we develop innovative ensemble learning methods to better detect the lateral movement attack and PBX fraud activities. Our contribution is threefold. First, we propose a supervised and an automatic semi-supervised approach based on ensemble learning to detect all the related activities to the lateral movement attack. Then, we present how to detect PBX fraud activities by developing approaches based on unsupervised learning coupled with ensemble learning. Finally, we propose a one class classification method coupled with ensemble learning that learns unsupervised representations to improve the detection rate of several anomaly detection problems.

Our experimental datasets, extracted from well-known institutions where the privacy and the confidentiality were ensured, support our contributions. In addition, real-life enterprise data, provided by POST Luxembourg, were extracted to address the problem of detecting PBX fraud activities. In this thesis, we provide the motivations of our anomaly detection research project, describe the theory employed to improve state-of-the-art approaches and quantitatively evaluate our methodologies.

Table of contents

List of figures

List of tables

List of Publications

1. **Georgios Kaiafas**, Georgios Varisteas, Sofiane Lagraa, Radu State, Cu D Nguyen, Thorsten Ries, Mohamed Ourdane. Detecting malicious authentication events trustfully. In the 2018 NOMS IEEE/IFIP Network Operations and Management Symposium. Best paper award

2. **Georgios Kaiafas**, Christian Hammerschmidt, Radu State, Cu D Nguyen, Thorsten Ries, Mohamed Ourdane. An Experimental Analysis of Fraud Detection Methods in Enterprise Telecommunication Data using Unsupervised Outlier Ensembles. In the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management.

3. Faouzi Amrouche, Sofiane Lagraa, **Georgios Kaiafas**, Radu State. Graph-based malicious login events investigation. In the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management. Best short paper award

4. **Georgios Kaiafas**, Christian Hammerschmidt, Sofiane Lagraa, Radu State. Auto Semi-supervised Outlier Detection for Malicious Authentication Events. In the 1st Workshop on Machine Learning for Cybersecurity (MLCS) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2019).

5. **Georgios Kaiafas**, Sofiane Lagraa, Radu State. One-class classification Ensembles with Unsupervised Representations to Detect Novelty. Under submission

Chapter 1

Introduction

According to Andrew Moore, former-Dean of the School of Computer Science at CMU, *"Artificial Intelligence is the science and engineering of making computers behave in ways that, until recently, we thought required human intelligence."* Siri and Google Assistant are two artificial intelligence–powered virtual assistants which learn from the interaction with the user and provide personalized user experience. They are probably the most iconic examples of artificial intelligence abilities of gadgets. Artificial intelligence contains many sub-fields, including: (i) Machine Learning, (ii) Computer Vision, (iii) Natural Language Processing (NLP). Computer vision algorithms interpret and understand the visual world. In particular, such algorithms give the ability to machines to accurately identify and classify objects, capture and interpret images or videos in real time, and also react to what they "see." NLP gives the ability to computers to analyze, understand and generate human language, including speech. Machine learning is a sub-field of AI that automatically learns and improves from experience without being explicitly programmed. According to Tom Michael Mitchell, former Chair of the Machine Learning Department at CMU, *"Machine Learning is the study of computer algorithms that allow computer programs to automatically improve through experience"*. Machine learning algorithms are inspired by neural networks, statistics, operations research and physics in order to find hidden patterns. The most distinguished subset of machine learning algorithms is the family of deep learning techniques that is steadily gaining popularity due to its achievements. The essence of deep learning is the neural network or artificial neuron [95]; an elementary unit receiving a weighted input. Deep learning is behind driverless cars, object detection in videos or images, language translation, song composition etc. A typical deep learning model is composed of different layers of neural networks where the first layer receives the input data and the last layer of neurons outputs the processed information. The intermediate or hidden layers perform nonlinear transformations of the inputs.

Machine learning techniques can be divided into the following broad categories: (i) supervised learning, (ii) semi-supervised learning, (iii) unsupervised learning, (iv) reinforcement learning. Learning with supervision means that someone can guide us towards to the right answer. Likewise, in the machine learning context, supervised learning methods require a full set of labeled data to train an algorithm to the correct answer. However, in real-world scenarios, obtaining fully labeled data sets is difficult because it is a time intensive task that requires huge manual work and significant domain knowledge. In addition, supervised learning approaches are mainly useful for classification problems and regression problems. On the other hand, unsupervised learning approaches are able to extract patterns from data without any supervision. In particular, unsupervised learning methods are mainly used for (i) cluster analysis, (ii) anomaly detection, (iii) dimensionality reduction, (iv) density estimation. It should be noted that, in an unsupervised learning setting there is no specific desired outcome or correct answer. Semi-supervised learning is halfway between supervised and unsupervised learning. It is suitable for the scenarios where a large amount of unlabeled data is available in conjunction with a small amount of labeled data. Semi-supervised learning can be based on either *transductive learning* or *inductive learning* [55]. Finally, reinforcement learning methods learn an optimal policy that maximizes a reward based on a set of actions, or decisions. Reinforcement learning methods develop intelligence by interacting with the environment and rely on the concept of trial and error. It is essentially inspired by the way humans or other intelligent beings learn. Recently, deep reinforcement learning [103, 255, 187], which refers to the combination of reinforcement learning with deep learning, is gaining popularity.

Ensemble learning has demonstrated a great success in the machine learning field. More precisely, ensemble learning techniques use multiple learners, which are usually called *base-learners*, to solve the same problem. The objective is to combine those learners to ultimately produce a meta-model that outperforms the individual models. In other words, the objective of ensemble learning is to improve the generalization ability of the base learners. Typically, an ensemble is constructed either in a parallel or sequential fashion. In a parallel ensemble, multiple base-learners are independently executed of one another whereas in a sequential ensemble there is a dependency between the base-learners in a sense that the first base learner influences the subsequent base-learner(s). A good ensemble, is composed of base learners that are as much accurate and diverse as possible [152]. Bagging [37] (Bootstrap Aggregation), Boosting [84] and Stacking [262] are three representative and effective examples of ensemble learning that can be applied to the problems of regression or classification. Boosting methods, such as AdaBoost [85], are based on improving the accuracy of a weak classifier by focusing on correcting instances that are not classified correctly by the previous classifiers. Weak classifier is a classifier that performs better than

random guessing. Bagging methods, such as Random Forest [38], are based on learners that suffer from high variance. The objective is to combine these learners by following the bootstrapping technique in order to reduce the total variance. Bootstrap samples are obtained by subsampling with replacement. Stacking is a two-phased ensemble learning paradigm where a number of first-level individual learners are employed. Afterwards, the output of those first-level learners is leveraged by a second-level learner which is called as meta-learner.

A lot have changed from the first computer worm attack in 1989, The Morris Worm [132], that distributed via the internet and infected around 6000 computer. However, the only thing that remained unchanged is that attacking tactics are evolving at a rapid pace without slowing down. From computer worms to large data breaches, attacks take any shape and size. In regards to the most sophisticated and hazardous cyber attacks, Advanced Persistent Threats (APTs) is the most representative example. In particular, a team (count tens or even hundreds of people) of highly-skilled intruders establish a long-term presence on the network of high-profile victims. The goal is to stealthily exfiltrate valuable and sensitive data. Hence, due to how well organized and sophisticated these attacks are, most of the security systems are not able to detect or prevent such type of attacks. APTs usually target physical critical systems and Stuxnet [130] is the most infamous attack that devastated Iran's nuclear program. The aforementioned facts show that no-one can feel totally protected against cyber attacks. This is also confirmed by the findings of a study conducted by Kaspersky in 2018. The findings of this study show that cyber-threats are considered as one of the top 3 developing risks in two years' time by almost half of all organisations. Additionally, in the last twelve months, 91% of organisations have experienced at least one attack most commonly in the form of malware. Moreover, Robert S. Mueller, III, former Director of the FBI made the famous quote: *"There are only two types of companies: Those that have been hacked and those that will be hacked"*. As such, adversaries are capable of finding their way to hack the assets of organizations or institutions or enterprises.

Telecommunication services go hand in hand with the internet and technology and this is also depicted in Fig. 1.1. As a result, nowadays there are numerous ways to communicate vocally that do not rely on the PSTN (Public Switched Telephone Network) such as Facebook chat, Skype, WeChat, WhatsApp, Zoom etc. Today's digital environment has reduced cost and increased availability of telecommunications equipment capable of hacking inter-carrier trust. As such, adversaries find fertile ground for fraud activities to gain financial benefit. According to Financial Times [83], the financial cost of telecommunication fraud reaches $17bn in revenue a year. Telecommunication fraud schemes rely more and more on exploiting the vulnerabilities of the internet. As such, cyber-telecommunication crime

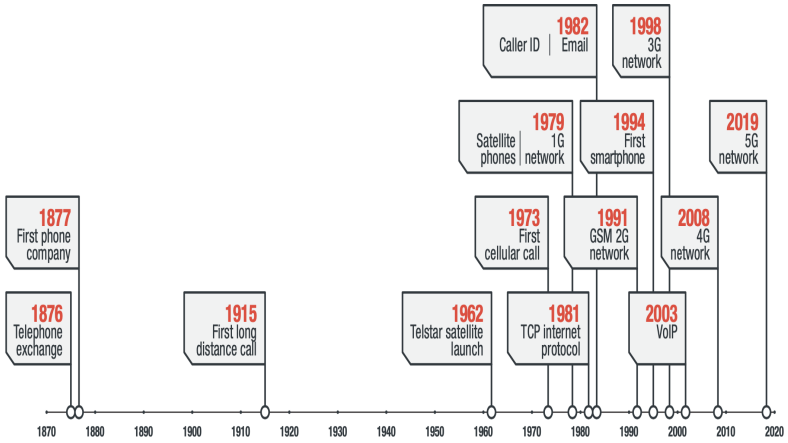

Fig. 1.1 Timeline of telecommunication evolution [153]

blooms and can be as effective as traditional cyber-crime. The authors in [82] provide a thorough analysis of the cyber-telecommunication fraud environment and discuss how challenging is for telecommunication carriers to defend against criminals. A global and unified threat intelligence that shares intelligence and techniques is crucial to fight the cyber-telecommunication crime that is a multi-billion dollar industry. Hence, it is critical not to be idle but to constantly enhance security against cyber and cyber-telco criminal activities.

In this hostile and constantly evolving cyber-criminal environment, the enterprises have to advance their detection mechanisms against cyber and cyber-telco criminal activities. The amount of such activities is steadily increasing at such levels that manual investigation is no longer a viable solution. In addition, it is vital for the survival of enterprises and organizations to detect cyber-criminals pro-actively instead of re-actively. In order to achieve it, enterprises can take advantage of the unparalleled technology advancements that give the opportunity to store and quickly analyze data. To give an example of these advancements we compare today's computers that crunch petaFLOPS (quadrillion FLOPS) with ENIAC, the first computer, that processed about 500 FLOPS (Floating Point Operations). Furthermore, enterprises can improve their detection mechanisms by leveraging the latest techniques originated from the intersection of computer science and statistics, such as machine learning (ML) and artificial intelligence (AI). These techniques are able to identify complex rules and patterns thus are suitable to expose the sophisticated behavior of criminals. To better understand the research problems of this thesis, we elaborate on these generic terms and the techniques related. Although there is no distinct boundary between the notion of AI and ML we try to amplify their differences and shed light in this novel field. Artificial intelligence and machine learning share the same configuration which is composed of a dataset used to employ an algorithm that is capable to find rules, patterns within this dataset.

1.1 Machine Learning for Anomaly Detection

It is natural to expect that anomaly detection techniques take advantage of AI and ML advancements to reach their objectives. In particular, anomaly detection methods rely on machine learning and artificial intelligence to identify unexpected and outlying behaviors that are in someone's interest. Detecting anomalies is crucial in many domains, including for example, fighting several types of crime such as financial crime, cyber crime, cyber-telecommunication crime, detecting tumor in image data, identifying pathologies in medical data, detecting faults in industrial systems and visual inspection using drones. The decision how to address an anomaly detection problem could be influenced by multiple factors but the most prominent is the ground truth availability. Hence, the large majority of anomaly detection approaches could be either supervised or unsupervised or semi-supervised. More specifically, supervised learning methods require fully labeled datasets in which the anomalous behavior is known in advance. Learning with supervision helps the algorithms find rules or patterns regarding how to distinguish normal and abnormal behavior. Supervised anomaly detection approaches are a special instance of the family of supervised algorithms due to the relatively few members of the anomalous class. Furthermore, unsupervised learning methods play a key role in the anomaly detection field. Unsupervised learning methods in contrast to supervised learning do not require any knowledge related to the anomalous behavior which makes such methods a natural approach to the problem of anomaly detection. Unsupervised learning methods can classified into the following groups: (i) proximity-based that model outliers as points which are isolated from the remaining data. LOF [39] (Local Outlier Factor) is one of the most popular algorithm of this group. (ii) statistical-based that model data using probability distributions. Extreme Value Analysis (EVA) [197] is one of the most popular algorithm of this group. (iii) subspace-based that find relevant lower projections of the data. Feature Bagging [159] is one of the most popular algorithm of this group. (iv) one-class classification-based. One-Class SVM [223] and SVDD [242] are two of the most popular algorithms of this group. Finally, semi-supervised learning approaches, that use both unlabeled data and labeled data [55], provide a viable solution for the problem of anomaly detection. More specifically, such methods aim at finding a description of the normal class and anomalies are exposed by assessing the divergence from the normal behavior. $SVDD_{neg}$ [243] is the most popular semi-supervised algorithm for anomaly detection.

Furthermore, ensemble learning has demonstrated success in the problem of anomaly detection by combining diverse results originating from different models. Ensemble learning methods not only collectively improve the overall performance of the final model but also alleviate the user from the subjectivity and ambiguity that profoundly influence the definition of anomalies. This ambiguity and subjectivity are depicted in each attempt to formally define

what is an anomaly [105, 28, 97]. Specifically, one of the first definitions of anomaly is given by Grubbs et al. in [97]: *"An outlying observation, or outlier, is one that appears to* ***deviate markedly*** *from other members of the sample in which it occurs"*. It is not difficult to observe that the adverb *"markedly"* carries a lot of subjectivity that is transferred to the anomaly detection algorithm. As such, anomaly detection algorithms are designed to be biased towards a part of the total anomalous truth. In other words, they are designed to partially expose outliers or anomalies. On the other hand, by combining diverse biases it is possible to produce a more effective final model compared to each individual model. The effect of combining diverse ensemble members is illustrated in Fig. 1.2 whereas Fig. 1.3 illustrates a counterexample.

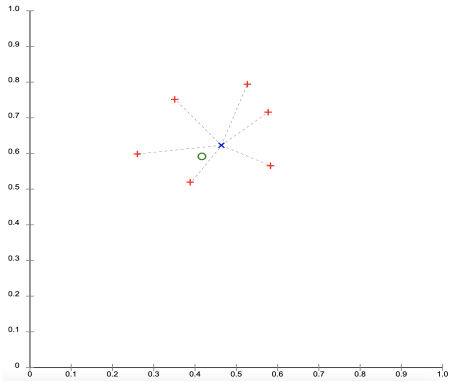

Fig. 1.2 Diverse outlier scores [279]. In red are coloured the ensemble members, in blue the final ensemble model, in green the ground truth

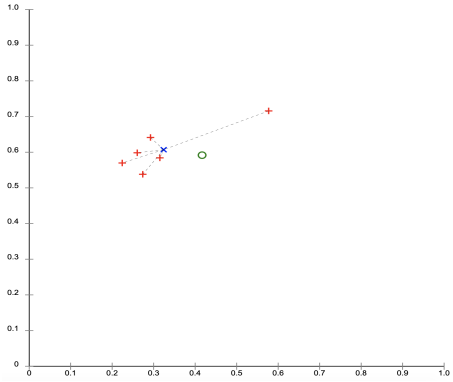

Fig. 1.3 Non-diverse outlier scores [279]. In red are coloured the ensemble members, in blue the final ensemble model, in green the ground truth

1.2 Research Problems

Throughout this chapter, we have discussed several challenges of the cyber-security and cyber-telecommunication domain. Existing defensive approaches against adversaries are not sufficient and they fail to effectively detect anomalous activities. Hence, there is plenty of room for innovation and improvement. Moreover, we shed some light on the differences between artificial intelligence and machine learning. In addition, basic anomaly detection and ensemble learning concepts were discussed. The motivation of this thesis is to propose innovative and novel machine learning methods to address the discussed challenges in the cyber-security and cyber-telecommunication domain. Novel solutions in real-world scenarios are introduced to increase the defensive barriers of enterprises and protect their Achilles heel against adversaries. More precisely, we take advantage of the merits of ensemble learning to

detect anomalous behavior in a robust and effective way. Therefore, in this thesis we develop methods that address the following research problems:

- We focus on the effective detection of a sophisticated cyber-attack, called Lateral Movement Attack, that is considered as Advanced Persistent Threat due to its characteristics. Currently, the largest body of research on detecting Lateral Movement Attack is restricted on identifying anomalous entities instead of identifying anomalous events. Hence, there is a need to provide actionable insights to analysts by answering questions related to when exactly and at which systems a malicious event happened. In addition, there is lack of research on ensemble learning methods for the Lateral Movement attack. As such, we interested in developing ensemble learning techniques for the problem of detecting the Lateral Movement Attack. In other words, *can we introduce innovative ensemble learning methods to effectively capture anomalous patterns of this particular cyber-attack?*

- We focus on the effective detection of telephony crime on the network of POST Luxembourg without any historical knowledge related to anomalous activities. Currently, there is a large scarcity in real-world datasets and limited academic work in telecommunication area that makes mandatory the need for novel fraud detection models. Additionally, the importance of developing robust and effective fraud detection approaches is high. For use in real-world business applications a method that can perform well on different types of data ensures that the introduced method will not impact that business in unexpected ways. As such, we are interested in detecting telecommunication fraud activities by introducing innovative anomaly detection models that do not need supervision and are based on well-established ensemble learning principles. In other words, *can we introduce innovative ensemble learning methods to effectively capture anomalous patterns of telephony crime?*

- We focus on the effective detection of novel anomalies by learning unsupervised representations. Novel anomalies correspond to previously unseen patterns of anomalies and their detection is critical in many real-life applications. The main areas of application could be the detection of faults in complex industrial systems, of structural damage, and of failure in electronic security systems, credit card fraud activities. Currently, the problem of learning unsupervised representations to detect novel anomalies has not been addressed properly. State-of-the-art methods require learning with full supervision to detect novel anomalous patterns. However, one-class classification methods have demonstrated success in the problem of detecting novel anomalous patterns. As such, we are interested in detecting novel anomalies by introducing an innovative

ensemble learning method based on one-class classification algorithms and unsupervised representations. In other words, *can we introduce innovative ensemble learning methods to effectively capture novel anomalous patterns regardless the application domain?*

1.3 Contributions

- In Chapter 2 of Part I we introduce a novel supervised ensemble learning method that is based on three individual classifiers and advanced graph-based feature engineering. It is able to effectively identify anomalous activities by achieving 0 false negative rate and on average a false positive rate of 0.0019. In Chapter 3 we introduce a sequential semi-supervised ensemble learning method that is developed using unsupervised outlier detection algorithms and one-class learners. It manages to improve the performance of the state-of-the-art. Our introduced methods not only are the first of their kind that follow ensemble learning techniques to address the problem of detecting the Lateral Movement attack but also provide actionable insights to the cyber-analysts.
- In Chapter 2 of Part II, we propose unsupervised outlier ensembles for the problem of detecting telecommunication fraud activities. In addition, we provide insights regarding the impact on selecting different ensemble components. Overall, our method effectively identifies the fraud activities happened in the network of the POST Luxembourg. Our introduced method attempts for the first time to detect real-life telecommunication fraud activities by developing ensemble learning approaches.
- In Chapter 1 of Part III we introduce a novel one-class classification ensemble framework that is built on unsupervised representations to improve baseline approaches with statistical significant results. In addition, our method, successfully extends existing supervised learning works that are developed using unsupervised representations. Our extensive experiments show that it is able to improve the detection rate in real-world scenarios especially when knowledge of the anomalous activity is scarce. Our introduced method is the first of its kind that effectively learns unsupervised representations and develops one-class classification ensembles in order to detect novel anomalies.

1.4 Thesis Structure

In the second chapter of Part I, we provide an introduction to the problem of anomaly detection, present the different settings of anomaly detection methods and discuss the main

challenges. Additionally, we elaborate on the ensemble learning for anomaly detection and present the state-of-the-art on this field. Ensemble learning for anomaly or outlier detection is increasing popularity due to its merits to overcome the typical difficulties of the anomaly analysis. A list of publicly available software is also provided. This thesis is divided into three disjoint parts that extensively discuss and provide innovative solutions to three different problems. The most significant commonality between these parts is that all the proposed solutions rely on Ensemble Learning foundations. The first part provides details related to the nature of a sophisticated and well-organized attack, namely Lateral Movement (LM) Attack. Additionally, it discusses the state-of-the-art detection methods of the LM attack. Moreover, this part is divided into two individual chapters in which we propose two ensemble learning based methods that are able to effectively detect the LM attack. The first method needs full supervision whereas the second is free of supervision. Both methods are evaluated on a freely available real-world dataset collected within Los Alamos National Laboratory's corporate, internal computer network. The objective of the conducted experiments is to demonstrate the effectiveness of our introduced ensembles in a cyber security application. The second part of this thesis, we present details regarding the telephony fraud ecosystem and also discuss the challenges and issues that fraud detection systems have to overcome in order to be effective. Additionally, we present current works that focus on fraud detection of telecommunication with the use of data mining techniques. In this part, we propose an unsupervised anomaly detection method that detects fraudulent Private Branch Exchange (PBX) phone calls made on the network of the largest provider in Luxembourg, POST Luxembourg. PBX is a critical enterprise technology that enables enterprise customers to manage their internal and external communication needs. The third part introduces an innovative one-class classification and ensemble method that addresses the novelty detection problem with the aid of unsupervised representations. More precisely, one-class classification learners are used to accommodate the scarcity of sufficient labelled training sets. By using unsupervised outlier scoring algorithms it is possible to learn unsupervised representations of the initial features of a dataset that are able to better expose outliers.

Chapter 2

Ensemble Learning for Anomaly Detection: Background and Related Work

Anomalies are also referred as outliers, abnormalities, or deviants [13] and one of the first definitions of outliers is given by Grubbs et al. in [97]: *"An outlying observation, or outlier, is one that appears to* ***deviate markedly*** *from other members of the sample in which it occurs"*. Hawkins et al. in [105] defined what an outlier is as it follows: *"An outlier is an observation which* ***deviates so much*** *from other observations as to arouse suspicions that it was generated by a different mechanism"*. Barnett et al. in [28] defined what an outlier is as it follows: *"An observation (or subset of observations) which* ***appears to be inconsistent*** *with the remainder of that set of data"*. It is apparent that subjectivity and ambiguity is present in each of the aforementioned definitions. As such, this subjectivity and ambiguity pose the most dominant challenge in the way to identify anomalous or unusual or unexpected or interesting or unlikely patterns. Therefore, before identifying data that exhibits such patterns it is crucial to determine what is an anomaly (outlier). The procedure of identifying data that exhibits anomalous or unexpected patterns is called anomaly (outlier) detection.

Fig. 2.1 illustrates how the different aspects of an anomaly detection problem influence the decision of the finding the appropriate algorithm for a given problem. In addition, Fig. 2.1 categorizes anomaly detection problems based on the (i) learning scenario, (ii) type of data, (iii) data structure, (iv) type of anomalies and highlights the different ways an anomaly detection problem could be approached. Detecting anomalies is very important in many domains, including for example, detecting financial or cyber crime, identifying pathologies in medical data, forensic applications or detecting faults in industrial systems. Furthermore, in this chapter we elaborate on well-established ensemble learning principles and techniques.

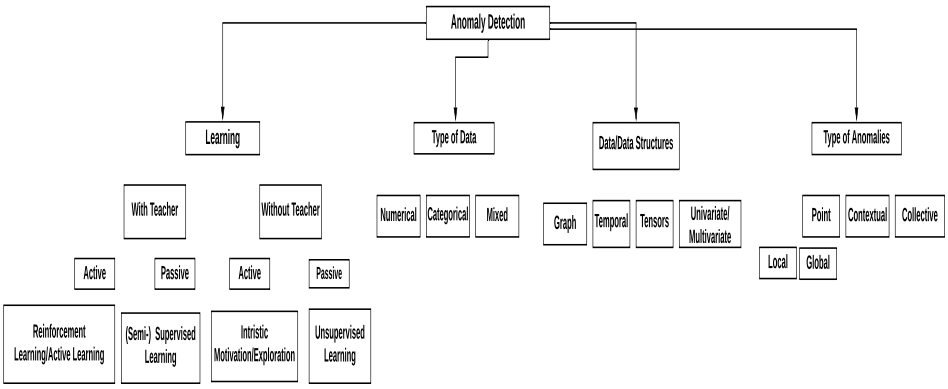

Fig. 2.1 A taxonomy of Anomaly Detection Approaches

More specifically, ensemble learning methods aim to alleviate the user from the subjectivity and ambiguity that profoundly influence the definition of anomalies. Finally, existing works related to ensemble learning methods for the problem of anomaly detection will be presented.

2.1 Anomaly Detection

As previously discussed, anomalies could be considered as data objects that do not conform to expected normal behavior. On the other hand, it is challenging due to several reasons not only to define a region that represents the normal behavior but also to assess which data objects belong to this region. Below, the major challenges are presented:

- Often there is no clear border that distinguishes normal and abnormal behavior thus is difficult to label observations that lie close to the boundary.
- There is a huge scarcity of sufficiently labeled data for evaluating of anomaly detection methods.
- The notion of an anomaly is not the same between different application domains and very often normal and/or anomalous behavior shifts. As such, the current behavior is not representative anymore.

Type of Anomalies

Hence, due to the aforementioned challenges (i) detecting anomalies is not a trivial task, (ii) most of the anomaly detection techniques are able to partially capture the anomalous patterns.

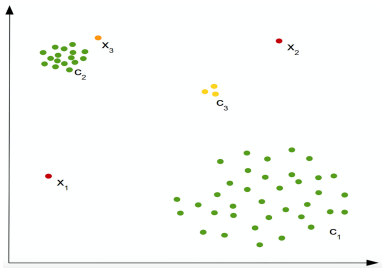

Fig. 2.2 A two-dimensional illustration of global anomalies (x_1, x_2), a local anomaly x_3 and a micro-cluster c_3. [94]

In addition, the nature of the desired anomaly to be detected could classify anomalies into different types. As such, the nature of anomalies influences the anomaly detection approach that will be followed. Chandola et al. in their survey paper [54] classify anomalies into the following three categories:

- Point Anomaly: An individual data object that is anomalous in regard to the rest of data. In addition, point anomalies can be divided into *Local* and *Global* anomalies as illustrated in Fig. 2.2. *Global* anomalies are considered anomalous relatively with all other data objects whereas *Local* with a subset of data objects.
- Contextual Anomaly: A data object that is considered as anomalous only in a specific context. It is also referred to as conditional anomaly [232]). In other words, the same data object may not be considered anomalous in a different context. Fig. 2.3 illustrates an example of time series data.
- Collective Anomaly: A subset or collection of data object is considered as anomalous but the individual data objects are not necessarily considered as anomalies. In particular, it is their collective occurrence that deviates significantly from the entire data set. Fig. 2.4 illustrates an example of collective anomaly. The highlighted region (grey colour) is considered as an anomaly because low values exist for exceptionally long time.

How to select the appropriate anomaly detection algorithm?

The phase of selecting the most suitable family of algorithms is crucial. It heavily depends on our existing knowledge related to the anomalous behavior. In addition, the objective of an anomaly detection task also influences the decision of selecting the appropriate family of algorithms. For instance, given the fact that there is a limited knowledge related to anomalous

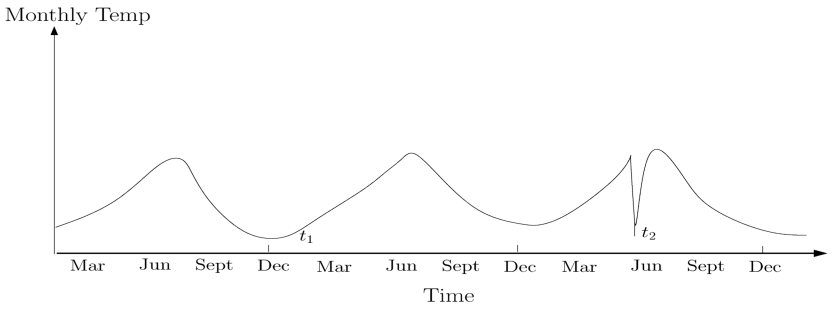

Fig. 2.3 A temperature time-series where t_2 is a contextual anomaly. Despite that temperature values at t_1 and t_2 are same, t_1 is not considered as an anomaly [54].

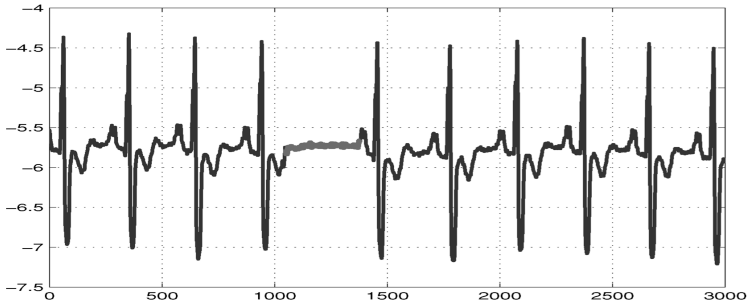

Fig. 2.4 Collective anomaly on data from a human electrocardiogram [93].

data objects are we interested in (i) *distinguishing between normal and anomalous classes based on existing knowledge* or (ii) *detecting previously unobserved anomalous patterns?*. The underlying objective of the first question is to learn a decision function to classify new and unseen data objects as anomalous or normal. The core element of this approach is that classified anomalies, can only help us to identify anomalies of a certain shape, colour or texture in the new and unseen data objects. In the second question, the underlying objective is to learn a model for the normal data objects and identify which new and unseen data differ in some respect from the normal model. The characteristic of such an approach is that anomalies of different shape, colour or texture in the new and unseen data objects could be identified. In this chapter in Sec. 2.3 we extensively discuss and elaborate on different learning scenarios.

2.2 High-dimensional data

High dimensional data are composed of a high number of features or independent variables. It is of high importance to detect anomalies in high-dimensional data because of the large number of applications such as banking fraud, network intrusion detection and financial

applications. However, outlier detection in high-dimensional data poses extra challenges for outlier detection with the most prominent, the 'curse of dimensionality'. The authors in [14, 283] discuss the effect of the 'curse of dimensionality' on exposing outliers in high-dimensional data. In particular, due to the 'curse of dimensionality' a high portion of irrelevant and noisy attributes may exist and as a result the contrast between inliers and outliers is diminished. Furthermore, the authors in [11, 113] investigate the impact of distance-based algorithms in scoring outliers and if the notion of neighborhood sustains in high dimensions.

Subspace outlier detection approaches constitute a possible solution to the problem of masking outliers by irrelevant dimensions. More specifically, such approaches assume that the outlying behavior is masked by full-dimensional analysis and outliers are often hidden in low-dimensional spaces (subspaces). As such, it might be more effective to search for lower projections in which the anomalous behavior is emphasized. Subspace based methods have demonstrated a huge success in anomaly detection [147, 190, 221, 131]. As such, by taking into consideration the behavior of the data in lower projections it could possible to design more effective algorithms. On the other hand, searching for relevant lower projections or subspaces could be computationally infeasible due to the high dimensionality of the data. In addition, as it pointed out in [10] selecting the "correct" or meaningful subspace is a process that leads to make mistakes. Hence, it is suggested to combine predictions from different subspaces in order to avoid making mistakes. Finally, we provide an example to illustrate how full-dimensional outlier detection analysis fails to effectively expose outliers. More specifically, in Fig. 2.5, four different 2-dimensional views of an artificially generated dataset are illustrated. Each of these views correspond to a disjoint set of dimensions. Data points 'A' and 'B' are exposed as outliers in the first and fourth view of the dataset respectively. However, neither of 'A' and 'B' are exposed as outliers in the second and third views of the data set. Therefore, the second and third views of the dataset are noisy and quite non-informative for exposing 'A' or 'B.' as outliers. As such, it is evident that (i) not all the views or lower projections are meaningful for the outlier detection analysis, (ii) outliers might be lost in a full-dimensional outlier detection analysis.

2.3 Learning Scenarios for Anomaly Detection

Anomaly detection approaches could be categorized into supervised and unsupervised based on whether the ground truth is known. The goal of unsupervised learning is to find interesting structure in the data whereas the goal of supervised learning is to learn a decision function given a set of labeled examples. In addition, except for supervised and unsupervised methods,

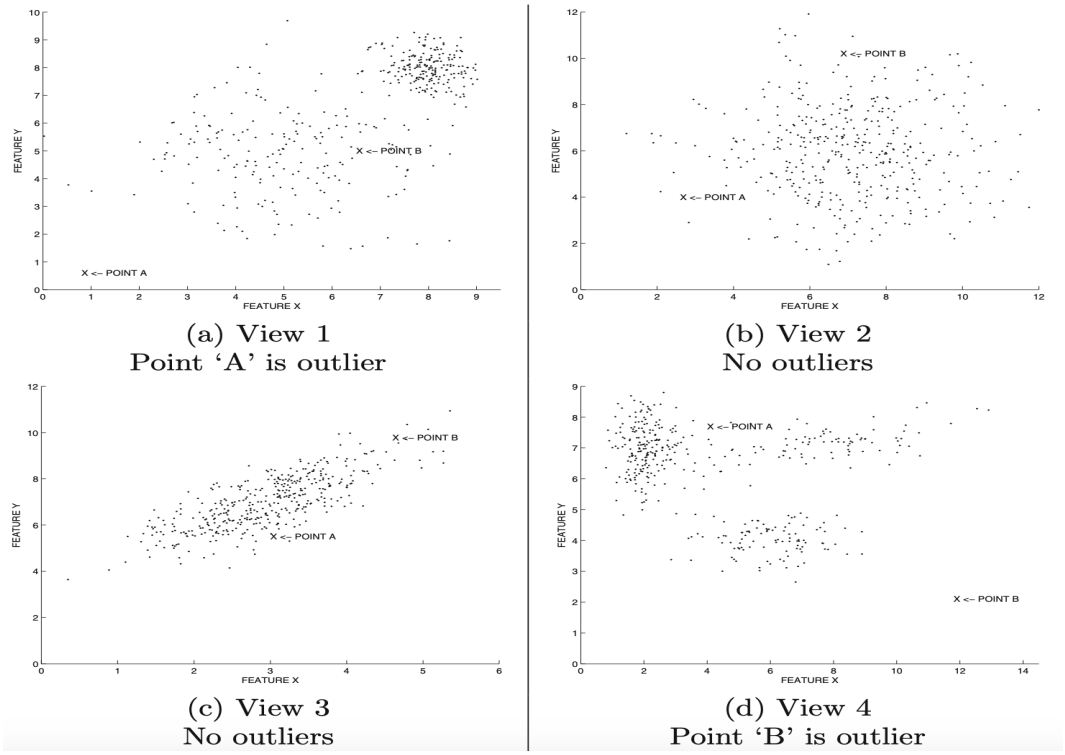

Fig. 2.5 Four different two-dimensional views of an artificially generated dataset [10]

semi-supervised learning approaches have been introduced to address the problem of anomaly detection. The typical scenario of semi-supervised learning is that normal data objects are available whereas outlier data objects are rare.

Imbalanced Data: Learning from imbalanced data is a common problem regardless the learning scenario (i.e. supervised, unsupervised, semi-supervised). According to [107]: *"Any dataset that exhibits an unequal distribution between its classes can be considered imbalanced."*. An imbalance ratio of 100:1, 1,000:1, and 10,000:1 is very common in two-class imbalanced datasets as well as anomaly detection problems. In other words, imbalanced datasets [45] could be used to evaluate anomaly detection methods. In this thesis we focus on three commonly used learning scenarios for anomaly detection: (i) *Supervised learning*, (ii) *Unsupervised Learning*, (iii) *Semi-supervised Learning*. Learning from imbalanced data with supervised approaches implies an imbalanced classification problem. On the other hand, learning from imbalanced data with unsupervised approaches implies a problem of data characterization. Following, we elaborate on the differences between these three large families of algorithms and we extensively discuss their applicability in the anomaly detection problem.

2.3.1 Ensemble Methods in Supervised Learning

Supervised Learning for Anomaly Detection

Supervised anomaly detection problems could be considered as very imbalanced classification problems due to the fact that the outlier class has relatively few members. Imbalanced classification problems are more than challenging because the majority of classifiers is designed assuming balanced classes; equal number of examples for each class. Imbalanced classification problems could affect either multi-class [60, 5, 234] or two-class classification scenarios. Henceforth, we are solely interested in the binary imbalanced learning problem which is closely related to the anomaly detection problem with numerous real-life applications [53, 207, 149].

Several studies [79, 29, 259] show that for a number of classification algorithms, balanced datasets improve their predictive performance compared to imbalanced datasets. As such, re-sampling methods are used to balance the skewed distribution of imbalanced data set in order to minimize its impact on the learning process. Re-balance is accomplished by (i) utilizing *oversampling* techniques that focus on artificially generating new data objects for the minority class. Various approaches are introduced to generate new data objects which span from methods that are based on nearest neighbors to deep learning methods [106, 156, 101, 56, 47, 165], (ii) utilizing *undersampling* techniques that focus on removing data objects from the majority class. Such techniques are mainly based on nearest neighbors methods [150, 175, 247, 102] (iii) combining *oversampling* and *undersampling* techniques. In addition, advanced re-sampling techniques [162] are proposed in order to preserve the structures of the classes and/or generate new data by sampling from the underlying distributions.

In addition, cost-sensitive learning methods [76] show success in addressing the imbalanced learning problem. More specifically, such methods incorporate varying costs (penalty) for each group of data objects to boost the performance of the learner. A higher cost is assigned to misclassified data objects that are less represented. Cost-sensitive learning approaches focus on minimizing the bias of a given learner towards majority groups. As such, they are a viable solution to imbalanced learning problems [48, 258, 245, 217]. Furthermore, several surveys [107, 108, 88, 173, 36] capture and provide a great overview of the recent advances in the imbalanced learning field.

Supervised Learning & Ensemble Learning for Anomaly Detection

Ensemble learning has become an effective and popular approach in addressing imbalanced learning problems [100, 143, 88]. Ensemble learning essentially is the combination of several models to ultimately produce a meta-model that outperforms the individual models.

Bagging (Bootstrap Aggregation) and Boosting are two examples of ensemble learning methods that can be applied to many statistical learning methods such as regression or classification [71, 152, 278]. Boosting methods, such as ADABOOST [85], are based on improving the accuracy of a weak classifier by focusing on correcting instances that are not classified correctly by the previous classifiers. Bagging methods, such as Random Forest [38], are based on learners that suffer from high variance. The objective is to combine these learners by following the bootstrapping technique in order to reduce the total variance.

Random Forest is an well established ensemble learning algorithm that grows unpruned classification or regression trees and uses random feature selection to improve performance. Random Forest is used in [57] address the problem of class imbalance. More specifically, the authors in [57] (i) introduce a cost-sensitive version of the Random Forest by penalizing misclassifications of the minority class and as a result, (ii) down-sample the majority class and growing trees on balanced datasets. BalanceCascade [168] is another example of supervised ensemble algorithms that its goal is to overcome inefficiency of the traditional random undersampling method. In particular, it develops an ensemble of classifiers by iteratively under-sampling the imbalanced dataset using an estimator. In addition, ensemble techniques are applied with the the SVM algorithm [7] to address the imbalanced problem. SVM is a non-parametric kernel-based algorithm that its objective is to construct a hyperplane in a high-dimensional space to maximize the separation margin between the support vectors and the hyperplane. However, the authors in [263] show that in imbalanced scenarios the decision boundary is largely biased toward the minority class. Hence, several works based on ensemble methods accompanied by re-sampling techniques (undersampling or oversampling) are introduced to minimize this bias [257, 129, 169].

2.3.2 Ensemble Methods in Unsupervised Learning

Unsupervised Learning for Anomaly Detection

Anomaly detection is inherently an unsupervised problem due to the fact that very often examples of outliers are not available. In an unsupervised setting, proximity-based approaches are the most established. The fundamental assumption of such approaches is that proximity of an outlier to its nearest neighbors is significantly different compared to proximity of an arbitrary data object to most of the other data objects in the data set. Proximity-based methods quantify the similarity between data objects using distance measures. As such, they are considered unsupervised approaches and as a result they do not leverage the ground truth in order to identify outliers. Furthermore, such methods could be classified into two types based on the type of output they produce. In particular, **labeling outlier detection methods**

output a binary vector that represents whether a given object is an outlier or not. **Scoring outlier detection methods** assign each object a score that express the degree of outlierness. Finally, proximity (similarity) can be defined in several ways and the most common are the following:

Cluster: Outlier detection and clustering problems have a very close relationship. In particular, the membership of a data point in any cluster is used to define proximity (similarity). More specifically, normal data points form dense clusters whereas outliers either do not belong to any clusters or form sparse clusters. The authors in [246] employ the well-established DBSCAN [80] algorithm with multiple parameters to detect anomalies in network traffic. In addition, the authors in [177] develop a three-phased method that uses the Affinity Propogation (AP) [86] clustering algorithm to detect anomalies in multi-view data; individual objects are described from several disjoint perspectives or views. In addition, the Global-Local Outlier Scores from Hierarchies method (GLOSH) outlier detection algorithm is a byproduct of the clustering algorithm HDBSCAN [43]. It is designed based on hierarchical density estimates computed by HDBSCAN.

Distance: Distance based outlier detection algorithms use the distance of a data point to its k-nearest neighbor to define proximity. Large k-nearest neighbor distances are indicative to expose outliers. On the other hand, often is computationally expensive to calculate the k-nearest neighbor distances of each data point. Different variants of distance based algorithms have been proposed in the literature. In particular, ODIN [104] is an algorithm that uses the number of reverse k-nearest neighbors to expose outliers. Furthermore, the authors in [206] use the distance of a point to its k_{th} nearest neighbors as a proximity measure.

Density: The density around a data point is compared with the density around its local neighbors. This comparison is called *relative density* of a data point and is used to define proximity. Density based algorithms assume that the density of a normal data point is similar to the density around its neighbors. On the other hand, this is not the case for outliers. The Local Outlier Factor (LOF) [39] method is a well-established algorithm that compares the local density of an object to the local densities of its neighbors in order to expose outliers. Several variants have been proposed that are similar to LOF such as Local Correlation Integral (LOCI) [196] and LDOF (Local Distance-based Outlier Factor) [272].

Furthermore, unsupervised anomaly detection is also present in the one-class classification (OCC) context. More specifically, OCC approaches for outlier detection belong to the broader one-class classification family of algorithms in which the objective is to learn a decision function that distinguishes between normal and unusual observations. In addition, OCC methods could be categorized into two broad categories (unsupervised and semi-supervised) based on the availability of outliers examples. In particular, *unsupervised*

one-class classification approaches do not have any mechanism to directly use label information learn the decision function. In addition, if one-class classifiers are able to use label information they are considered *semi-supervised learning approaches*. We provide details regarding semi-supervised OCC in the next section. Finally, one-class learners could be either density methods or boundary methods or reconstruction methods. Boundary methods define a boundary around data in order to accept as inliers (normal) observations that fall within the boundary. Observations that fall outside of the boundary are considered as outliers. The Support Vector Data Description (SVDD) method [242] is one of the most popular boundary based OCC approach. PCA [125] and Autoencoder [**?**] are two of the most popular OCC methods that expose outliers by assessing the reconstruction error of the input space. Density based OCC methods use training data to estimate the p.d.f of the data and new observations are classified using this p.d.f.. Gaussian density is a commonly used OCC that is based on density [32].

Unsupervised Learning & Ensemble Learning for Anomaly Detection

Ensemble learning approaches for unsupervised learning are more challenging to be designed. However, the authors in [10, 12] show that outlier analysis and classification share similar theoretical foundations. Aggarwal et al. in [12] extensively discuss the bias-variance problem in the unsupervised outlier detection context. In addition they point out that even though the ground truth is unknown in an unsupervised setup, bias and variance can still be defined. They discuss the effect of bias in constructing outlier ensembles, how bias and variance could be minimized and also they review the feature bagging [159] technique and it variants. Zimek et al. in [279] transfer the accuracy and diversity principles of supervised ensemble construction into the unsupervised ensemble construction. They highlight the fact that, in an unsupervised setup, it is almost impossible to measure accuracy during the learning phase in contrast to diversity that can be measured.

Often, outlier detection algorithms score data objects according to their outlierness. As discussed previously, diversity of outlier detection models is a core ingredient of constructing good outlier ensembles. As such, in Fig. 2.6 and Fig. 2.7 the effect of diverse outlier scores in constructing outlier ensembles is illustrated. Our example is constrained in a two-dimensional world where each outlier detection algorithm produce a two-dimensional outlier score vector. The corresponding two-dimensional score vectors are coloured in red while in green is coloured the ground truth. Finally, in blue is coloured the outlier ensemble (two-dimensional score vector) that is constructed by calculating the average of the six individual outlier two-dimensional score vectors. Fig. 2.7 illustrates a scenario where diversity is limited or even missing while in Fig. 2.6 there is diversity in some extend. It should be noted that

in both scenarios all the individual outlier scores are quite accurate but when diversity is missing the accuracy of the constructed ensemble is negatively affected. In addition, it can be seen that in Fig. 2.7 all individual outlier detection results make the error and they form a tight cluster. On the other hand, in Fig. 2.6 the outlier detection results make different errors.

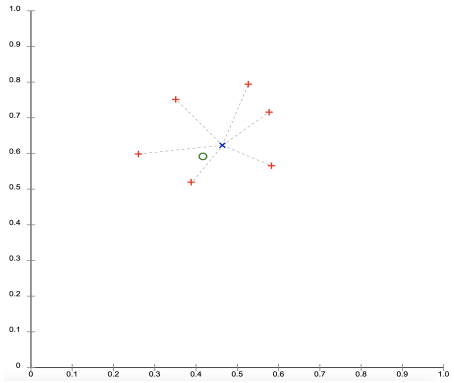

Fig. 2.6 Diverse outlier scores [279]

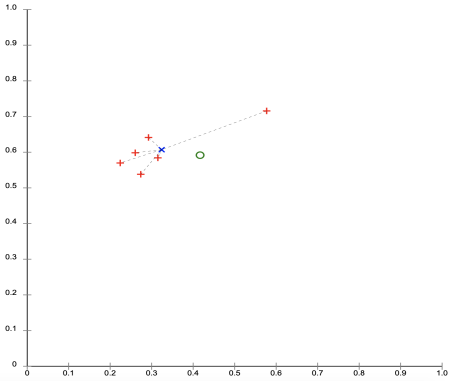

Fig. 2.7 Non-diverse outlier scores [279]

The authors in [159] introduce for the first time an outlier ensemble method method that is inspired by the well-established bagging technique. High diversity increases the chance that the models will make different errors. As such, the authors in [281], induce diversity by proposing an outlier detection method based on the subsampling technique. The authors [198] induce diversity by using both techniques; subsampling and bagging. Randomness is frequently used in order to induce diversity and construct good outlier ensembles. Such ensemble methods use arbitrary random data projections in order to obtain accurate results. An example of such ensemble methods is the isolation forest algorithm [167] which uses a randomized process to expose outliers in local subspaces of low dimensionality. Additionally, a randomized version of the hashing technique is the basis of ensemble subspace-histogram approaches [221, 222] that build histograms on random data subspaces and also scale linearly. LODA (Lightweight On-line Detector of Anomalies) [200] is another example of using random projections to build outlier ensembles.

Unsupervised combination of individual models into an ensemble

Combining different outlier scoring algorithms requires the same meaning of outlierness between the combined models. Often, this is not the case because outlier scoring algorithms produce outlier scores that differ in their semantics and meaning. As such, re-scaling outlier scores provides a solution to make feasible the combination of results from different models. Also, re-scaling often increases the contrast between inlier scores and outlier scores to better expose outliers. The authors in [146] show that re-scaling outlier scores increase the contrast

between outlier and inlier scores. In addition, the authors discuss the differences between normalization and standardization and they propose scaling methods for several algorithms in order to increase the contrast between inliers and outliers. Additionally, they introduce a novel procedure for normalizing outlier scores by using the gaussian and gamma distribution.

Recently, the authors in [44] inspired by supervised boosting methods and introduced the *BootSelect* strategy to construct unsupervised outlier ensembles. This is the first attempt to transfer supervised boosting methods in the unsupervised world. Their experiments showed that they outperform existing model selection methods. The authors in [224] introduce an unsupervised greedy heuristic to optimize diversity in outlier ensembles. Furthermore, the authors in [209] propose an unsupervised method that builds an ensemble by examining which results to combine from several different methods. In addition, majority voting or weighted majority voting techniques [151] could be used to combine the predictions of several individual models. Another example of unsupervised ensembles is proposed by the authors in [59]. They build an ensemble using the autoencoder method and obtain high quality results by allowing each autoencoder to overfit and inducing randomness. The experimental setting demonstrate that the introduced ensemble gains in efficiency.

Supervised and Unsupervised Learning for Anomaly Detection

Furthermore, the idea of leveraging the best of the supervised and unsupervised world inspired the authors in [183] who first presented such an approach. In particular, outlier detection algorithms that are able to score data according to their outlierness were used. The produced outlier scores were concatenated with the original features to train a supervised outlier detection method. More specifically, they used the logistic regression model with the ℓ_2 penalty trained with re-sampling and bagging to deal with class imbalance. They improved detection performance by comparing their proposed method against the original feature space. This work inspired others, but all of them addressed the supervised outlier detection problem. The authors in [184] proposed an extension with a feature selection method based on a finite budget related to the number of computations. The authors in [274] inspired by [184] and applied the XGBoost [62] classifier to deal with imbalanced data. Finally an application in credit card fraud detection [49] is introduced. Overall, all existing works tackle the supervised problem either with re-sampling techniques accompanied by penalization strategies or employing ensemble classifiers.

2.3.3 Ensemble Methods in Semi-Supervised Learning

Semi-supervised learning approaches use both unlabeled data and labeled data [55]. In the anomaly detection context semi-supervised methods aim at finding a compact description of the normal class. As such, they anomalies are detected by assessing the divergence from the normal behavior. The authors in [114] discuss the three fundamental approaches to the problem of outlier detection (supervised, unsupervised, semi-supervised) and they name the semi-supervised methods as *novelty detection methods*. As such novelty detection and semi-supervised learning are inextricably linked.

Novelty detection aims to identify data objects that are not consistent with normal expectations. Typically, novelty detection methods include a training phase where the normal behavior is learnt. Afterwards, test data that diverge in some respect from the normal behavior are considered as novelties. Novelty detection could be find in the literature as anomaly detection, outlier detection, concept learning, one-class classification, single-class classification, or data description [63, 211, 123, 96, 223, 54]. In addition, the problem of novelty detection is very common in real-world scenarios such as machine diagnostics [243], faults and failure detection in industrial systems [240] or video surveillance [178, 70].

Despite the fact that using label information improves the performance of the anomaly detection task, very few approaches are developed on a semi-supervised setting. Research on semi-supervised ensemble learning is almost solely relayed to the classification problem [142, 171, 271, 228, 267, 238]. Such methods, assume that similar data objects belong to the class [55] but in the anomaly detection context this assumption holds only for the normal class; anomalies are not necessarily similar to one another.

$SVDD_{neg}$ [243] is one of the well-established semi-supervised method for anomaly detection namely. It is an extension of the unsupervised SVDD [243] (Support Vector Data Description) algorithm where anomalies are used to tighten the data description. The SVDD algorithms has inspired thousands of works to either propose an extension or address real-world problems [160, 191, 161, 58].In Fig. 2.8 is illustrated how different parameters of the SVDD algorithm shape different data descriptions and as a consequence the performance is affected. Recently, the authors in [212] extended the SVDD algorithm to the semi-supervised setting by developing a deep neural network method to detect anomalies. They use the backpropagation method to optimize their method and the evaluation is solely on image data i.e. MNIST and CIFAR-10. Furthermore, the authors in [96] propose a semi-supervised method anomaly detection method that extends the SVDD algorithm in order to leverage unlabeled and labeled data. Methods that are originating from the unsupervised world are appropriate for identifying new and unknown anomalies in contrast to those that are originating from the supervised world. In addition, semi-supervised learning effectively

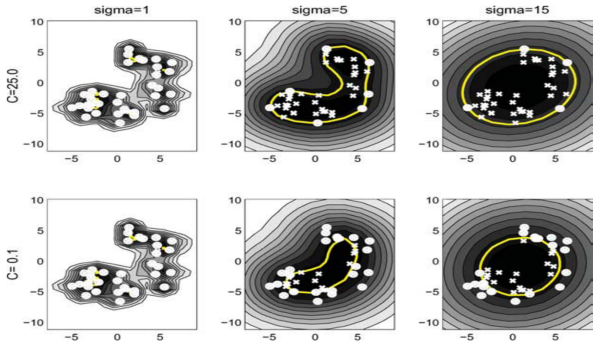

Fig. 2.8 Data description trained with Gaussian kernel with different widths and different C values. Support vectors are indicated by the solid circles whereas the solid white line is the description boundary. [243]

addresses the problem of novelty detection. The authors in [33] reduce the problem of semi-supervised anomaly detection to semi-supervised classification with a user-specified constraint α which represents the false positive rate. There is an increased interest in the semi-supervised setting for the problem of novelty detection. Several studies [75, 192, 15, 67, 68] are developed which they span from the adversarial training technique and VAE (Variational Autoencoders) [138] to shallow methods such as SVM.

.

2.4 Evaluation Measures

Anomaly or outlier detection research mainly focus on developing new methods and computationally improving these methods. However, it is crucial not to neglect the importance of evaluating the performance of existing methods in a meaningful way. In addition, it is equally important to fairly compare different methods and assess their similarity in order to ultimately select the best method. Outlier detection is an inherently highly imbalanced problem. As such, the evaluation measures of outlier detection methods should take into consideration the class imbalance problem.

The most popular and meaningful evaluation measure of anomaly detection methods is based on the Receiver Operating Characteristic (ROC) curve. ROC nicely addresses the typical problem of imbalance of class sizes; the amount of outliers (positive class) is excessively low compared top inliers (negative class). More specifically, the ROC curve is obtained by plotting the true positive rate versus the false positive rate. The true positive rate represents the proportion of outliers correctly identified whereas the false positive rate represents the proportion of outliers that falsely identified as outliers. Comparisons between

different curves is challenging thus a summarization of the ROC curve into a single value is preferred. More specifically, the area under the ROC curve (ROC AUC) summarizes a ROC curve and ranged between 0 and 1. A ROC AUC value of 1 is the corresponds to the perfect ranking whereas a ROC AUC value of 0 corresponds to the worst ranking.

Often the user of an anomaly detection method is only interested in a small subset consisting of the *n* top-ranked objects or top-n outliers. Several evaluation measures have been proposed in order to assess the performance of outlier detection methods based on the delivered top-n outliers. As such, it is natural to calculate the precision of a method at the top-ranked objects. More specifically, precision at n (P@n) [66] evaluation measure uses the correctly identified outliers results in the top-ranked objects. In a similar fashion, it is natural to calculate the recall of a method at the top-ranked objects. More specifically, recall at n (R@n) [13] evaluation measure is defined as the proportion of actual outliers that are correctly detected. In addition, by choosing n = |O| (n is equal to the number of actual outliers) the R-Precision measure [66] is calculated. This measure assumes that the number of actual outliers is known in advance, which is not always the case.

It should be noted that, the aforementioned measures require knowledge of the ground truth; labels identifying outliers and inliers. Often, in real-world anomaly detection problems labelled examples are missing. As such, measures that do not rely only on labeled knowledge would be more useful. These measures are called *internal evaluation measures* and are well established in unsupervised cluster analysis [170, 210, 189]. On the other hand, in outlier analysis there is huge scarcity of such internal measures. In the literature the only available internal evaluation measure is proposed by the authors in [179] but is considered as computationally expensive in multiple studies [45, 280].

2.5 Other Methods & Publicly Available Software

Deep learning methods is a family of machine learning algorithms that has demonstrated a great success in the task of anomaly detection. Several surveys [51, 154, 166, 27, 139] have been published that provide an overview of deep learning techniques for anomaly detection. Different architectures of the deep learning models have been introduced to detect anomalies in several applications. More specifically, Chalapathy et al. in their survey paper [51] review proposed deep learning architectures for anomaly detection and propose a categorization based on the input data. Table 2.1 provide details regarding this categorization.

Amongst the deep learning family of algorithms, the auto-encoder (AE) method has been quite effective in the task of anomaly detection. This is confirmed by the extensive body of research works [20, 193, 35, 270, 186, 24, 218, 277]. Auto-encoder was firstly introduced by

Table 2.1 Proposed deep anomaly detection model architectures in literature categorized based on the input data [51]. AE: Auto Encoder CNN: Convolution Neural Network, LSTM: Long Short Term Memory, RNN: Recurrent Neural Network

Type of Data	**Application**	**Deep Learning Model**
Sequential	Video, Speech, Time Series	CNN, LSTM, RNN
Non-Sequential	Image, Sensor, Other	CNN, AE and its variants

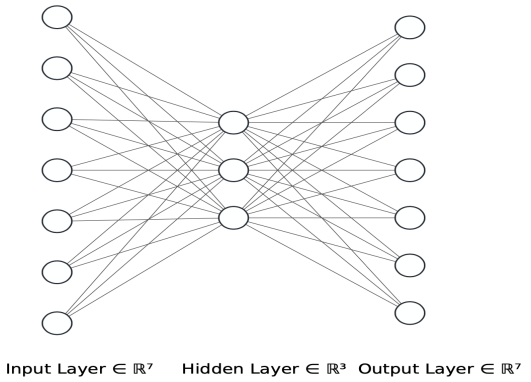

Fig. 2.9 An auto-encoder with one hidden layer.

the authors in [214] and is an unsupervised method that attempts to copy its original input x to its output. By assessing the magnitude of the reconstruction error of x (difference between the network's input and output) we are able to detect anomalies. In addition, auto-encoder is able to find linear or non-linear compressed representations of the original input x that could be used either for dimensionality reduction or feature learning. In Fig. 2.9 an auto-encoder method is illustrated; the input and output layers must be composed of the same number of nodes.

In Table 2.2 a summary of the publicly available software dedicated to anomaly detection approaches is presented. We categorize software based on the programming language and the capability of constructing ensemble methods. The large majority of the available software is written in Python.

Table 2.2 Summary of the publicly available software

Name	Related Research	Language	Ensemble Methods
ELKI [1]	Schubert and Zimek [225]	JAVA	Yes
pyod [2]	Zhao et al. [276]	Python	Yes
SUOD [3]	Zhao et al. [273]	Python	Yes
imbalanced-learn [4]	Guillaume et al. [162]	Python	Yes
Deep One-Class Classification [5]	Ruff et al. [213]	Pytorch	No
One Class Neural Network [6]	Chalapathy et al. [52]	Keras-Tensorflow	No
datastream.io [7]	-	Python	No

2.6 Conclusion

Ensemble learning has shown success in the problem of anomaly detection. In particular, several works successfully employed ensemble learning principles to outperform the individual ensemble members. In addition, regardless the learning scenario, i.e. unsupervised learning or supervised learning, these works could be divided into two broad categories: (i) a single anomaly detection algorithm is used in conjunction with a method like feature bagging and/or subsampling, (ii) multiple anomaly detection algorithms are combined to induce greater diversity. However, selecting the most suitable category for a given problem is data and application dependent.

In the next chapters, we deal with two applications of anomaly detection: (i) the detection of sophisticated cyber-attack, (ii) the detection of telecommunication fraud. More specifically, we introduce novel anomaly detection ensembles that span from supervised learning to unsupervised to effectively detect anomalous behavior. Well-established theory of ensemble learning [279, 10] is followed to develop scientifically rigid methods. Finally, we demonstrate the improvement of ensemble learning by experimentally evaluating our proposed methods.

Part I

Detection of Lateral Movement Attack

Chapter 1

Background and Related Work

According to Cisco [65]: *"Cyber-attack is a malicious and deliberate attempt by an individual or organization to breach the information system of another individual or organization"*. Often, the attacker seeks some type of benefit from disrupting the victim's network. In addition, the most common types of cyber-attacks are: (i) Malware, (ii) Phishing, (iii) Man-in-the-Middle, (iv) Denial-of-service, (v) SQL injection, (vi) Zero-day exploit, (vii) DNS Tunneling. We refer to [65] for extensive details.

In this chapter, we elaborate on an attack namely Lateral Movement which is a well-orchestrated and sophisticated attack. In addition we provide thorough details related to how this attack is performed and also discuss possible detection methods.

1.1 APTs and Lateral Movement Attack

Advanced Persistent Threats are the most sophisticated and hazardous cyber attacks. In particular, a team (count tens or even hundreds of people) of highly-skilled intruders establish a long-term presence on the network of high-profile victims. The goal is to stealthily exfiltrate valuable and sensitive data. Hence, due to how well organized and sophisticated these attacks are, most of the security systems are not able to detect or prevent such type of attacks. APTs usually target physical critical systems and Stuxnet [130] is the most infamous attack that devastated Iran's nuclear program. A report from Fireeye [50] summarizes the key findings of advanced persistent attacks:

- Intruders had established presence on victims' network for 205 days before they were discovered.
- 69% of victims detect ongoing attacks only with the help of a third party.

- Over the last year, intruders have introduced new stealthy lateral movements to stay undetected on victim's network.

Ussath et al. in [253] analyze the techniques and methods of 22 different recent APT incidents and determine common methods that are used within the different phases of APT attacks. They conclude that a common strategy of the intruders is to dump credentials to move laterally through the network. As such, the attackers are able to hide between legitimate traffic and activities and this fact makes the detection difficult while it allows them to bypass existing security systems.

Lateral Movement (LM) attack is a well-known and sophisticated cyber-attack that belongs to the category of APTs. Attackers maintain unauthorized access to a network for a long period of time without being undetected. The most common method to perform the initial compromise and ultimately cross the network perimeter border is achieved via a social engineering attack [176, 122, 41]. Upon compromising an initial computer of the organization, the adversaries use lateral movement techniques to access other hosts and mine sensitive resources. The ultimate goal of a LM attack is conquering the Domain Controller because it provides full control of the network.

Fig. 1.1 presents the typical scenario of an APT with lateral movement. LM is is a 2-step attack where at the first step the attackers capture credentials from a source host and at the second step they use stolen credentials to access another host or resources. Throughout this chapter we refer to lateral movement attacks as unauthorized connections from a source host to a targeted host using valid stolen credentials of an account (i.e., user or service account).

Soria et al. in [233] discuss possible ways to detect a lateral movement attack. They mention that there is no difference at protocol level between a legitimate connection and a connection with stolen credentials using a hacking technique. Pass-the-hash and Pass-the-ticket are two the most well known techniques (see [233] for details). As such, a LM attack does not exploit weaknesses at a protocol level but create anomalies at a behavior level. More specifically, a simple indicator of a potential lateral movement could be when a domain admin account is supposed to be used only from a specific workstation but eventually it is used from another workstation.

1.1.1 Techniques to perform a Lateral Movement Attack

This section discusses the most commonly used techniques to move laterally within a network. Lateral movement techniques are not lacking in number or diversity and the authors in [1] list and describe different techniques and mitigation strategies. These techniques may be different, but there is a clear pattern regarding the paths they follow to move within a network.

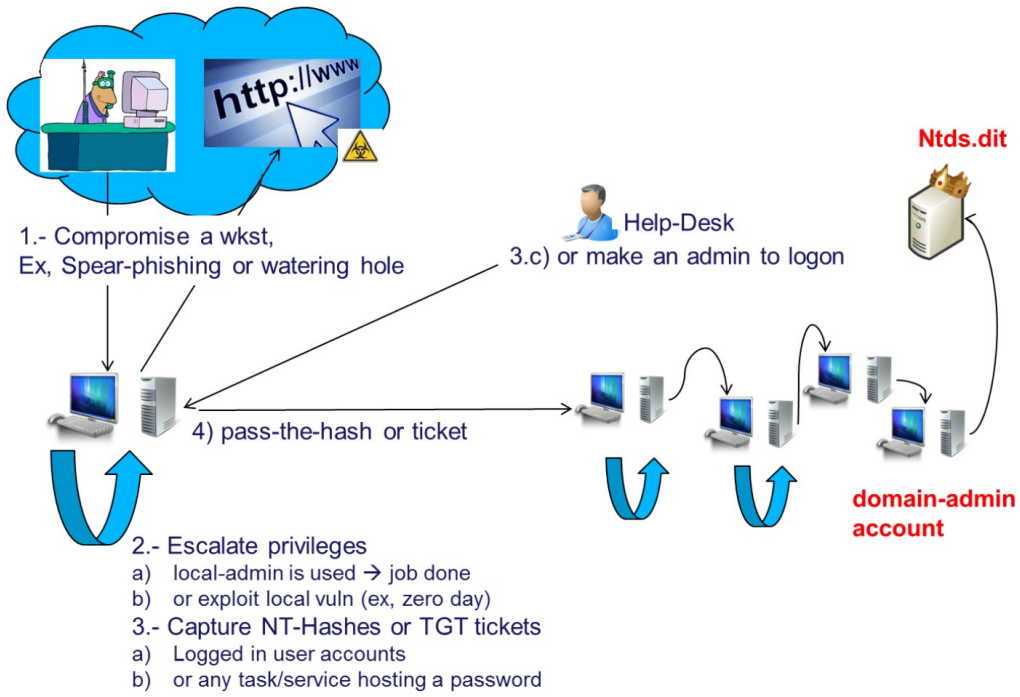

Fig. 1.1 Typical lateral movements in case of APTs [233]

It should be noted that, the lateral movement attack is typically executed behind the security network boundaries of an organisation.

Logon scripts: Windows operating systems permit logon scripts to be run whenever someone logs into a system. These scripts can be used to execute other programs or send information to an internal logging server.

Pass the Hash: This technique belongs to credential theft attacks and is a two-step process in which an attacker steals credentials and then uses those credentials to get access to other computers. In particular, the attacker dumps the hash of a compromised system to acquire all stored account credentials. Afterwards, the attacker uses the stolen credentials to move laterally on the network by acquiring account credentials close to domain controller. The risk associated with a Pass-the-hash attack varies depending on if an adversary manages to get administrative privileges on a compromised system. Recently, Microsoft published an article [126] to propose possible solutions regarding how to mitigate the effect of Pass-the-Hash attacks.

Remote Desktop Protocol: Microsoft designed the Remote Desktop Protocol (RDP) [2] to provide remote display and input capabilities. Moreover, Remote Desktop Service (RDS) is a Windows service that implements RDP. Usually, adversaries expand access by connecting to a remote system over RDP/RDS and they manage to get access to other accounts.

1.2 Evidence of an Attack

Attackers' actions can be detected by analyzing trace evidences in system logs. The most important data sources (evidences) available to an analyst are the following:

- **Event logs:** Operating systems record important software and hardware events from various sources and stores them in a single collection called an *event log*. Investigating security incidences heavily rely on event logs which is a critical resource [22]. Recently, Windows introduced the Sysmon [3] tool that offers an advanced log system.
- **Flow data:** Flow data are collected from IP network traffic as it enters or exits an interface. NetFlow is the most popular flow format.
- **Domain Name System (DNS) logs:** DNS translates domain names to IP addresses. However, several types of DNS-based attacks can be performed such as: (i) Domain hijacking, (ii) Domain flood attack (iii) Distributed Reflection Denial of Service (iv) DNS tunneling (v) DNS spoofing.
- **Web logs** Web traffic is typically allowed through firewalls. This makes web traffic attractive for attackers to hijack communication by installing malware on connected machines.

1.3 Related Work

In this section, we discuss existing works related to anomaly detection methods for the detection of the lateral movement attack. Existing works can be divided into three categories namely Machine Learning-based Anomaly Detection, Statistical-based Anomaly Detection and Graph-based Anomaly Detection.

1.3.1 Machine Learning-based Anomaly Detection

Over the last years machine learning and data mining techniques have successfully addressed anomaly detection problems [54, 283, 280, 51, 9]. As such, researchers have developed machine learning based detectors to address the problem of lateral movement detection or malicious authentication actions detection. The primary focus of the following works is the detection of unauthorized connections (authentications) from a source host to a targeted host using valid stolen credentials of an account.

Siadati et al. in [229] propose a method based on market basket analysis to uncover associations between items. In particular, login patterns are extracted from user authentication

histories in order to learn behavioral patterns and use these patterns to identify suspicious behavior. Their proposed method based on association rules is scalable for large datasets and consists of two classifiers, an exact matching classifier and a pattern matching classifier. Their system yields 82% recall and 99.7% precision in detecting malicious logons.

Siadati et al. in [230] propose a visualization tool, APT-Hunter that integrates security analysts' knowledge into the detection system. In other words, this detector does not enhance the existing knowledge of the cyber analysts via informing them for novel anomalous patterns but instead integrates known rules of anomalous actions. A tool that enables the analysts to enhance their existing knowledge regarding the anomalous patterns is of paramount importance. APT-Hunter is basically composed of (i) a login processor and (ii) an aggregator and pattern matcher. The authors report that after providing a 30 minutes training on APT-Hunter two participants are able to detect 349 out of 749 (46.5%) read team flagged malicious logins. However, they do not mention how much time those two users needed to find those anomalies

Chen et al. in [61] propose a a semi-supervised learning detector that is based on network embeddings. In particular, they construct a host communication graph from a variety of data. Then they learn features from the graph using network embedding methods and keep the most informative. Finally, autoencoders are used to reduce the dimensionality of the dataset and learn more informative features. Overall, Chen et al. demonstrate accuracy of 99.9% and precision of 91.3% in a balanced dataset which is not always the case in real-world problems.

Holt et al. in [116] introduce a method based on autoencoders with deep architectures. They test shallow and deep autoencoder architectures composed of different number of layers. They perform feature engineering by transforming input data into conditional probabilities of the authentication events in the network in order to improve performance. Their comparative analysis is very limited, composed of only two previous works. As such, despite the fact that their results are promising we can not consider them as strong.

Bohara et al. in [34] propose an unsupervised approach to detect malicious LM. In particular, they develop two unsupervised approaches on different data sources to combine their outputs and ultimately generate the final set of compromised hosts. The first detector is based on PCA and Kmeans and the second on PCA and extreme value analysis. They evaluate their approach by injecting artificial attacks into the LANL dataset. However, these simulated attacks may be subjective and not represent real attacks. Finally, the evaluation of this approach demonstrates the effectiveness of the introduced method on detecting long chains of infected hosts. In this thesis, we are interested in discovering infections composed of one malicious login.

Bai in [26] highlights the limitations of two publicly available authentication datasets from Los Alamos National Laboratory [252, 134]. In addition, he proposes an anomaly based method to detect malicious Remote Desktop Protocol (RDP) sessions. RDP is designed by Microsoft to provide remote display and input capabilities, while Remote Desktop Service (RDS) is a native service on Microsoft Windows platform that implements RDP [2]. Distinguishing between legitimate and malicious RDS use is challenging and thus attackers take advantage of this fact. As such, RDP is a primary tool used by attackers during a lateral movement attack. Bai investigates various supervised ML techniques and constructs ensembles from these models. The conducted evaluation and comparative analysis shows that a stand-alone model (LogitBoost [87]) performs the best in classifying RDP sessions and outperforms a state-of-the-art model in detecting malicious authentication events.

Bian et al. in [31] develop a supervised approach to early detect LM using the LANL dataset. In particular, they are inspired by our work [127] to (i) employ several supervised learning models and ultimately they develop an ensemble, (ii) utilize advanced feature engineering based on graphs. Finally, they compare their approach with the state-of-the-art and demonstrate improvement. In Chapter 2 we elaborate on our supervised method that inspired Bian et al..

1.3.2 Statistical-based Anomaly Detection

Heard et al. in [109] construct a directed authentication graph (V, E) where an edge $e \in E$ represents the presence of a directed connection from source computer x to destination computer y. Additionally, vertices represent destination computers and for each $y \in V$ a separate statistical model will be constructed for the identities of the sequence of source computers $x_1, x_2, ...x_n$ that connect to y as a destination. In particular, the Dirichlet process is used to model the distribution of source connections for each of the destination systems. Anomalies are found based on degree distributions of the source and destination computers. Dirichlet process [244] is a stochastic process that assumes that interchangeable sequences of $x_1, x_2, ...x_n$ which is not always the case in a lateral movement attack; the order of the sequence is vital to trace back the attack. The typical scenario of anomaly that this method tries to capture is a small number of source computers tries to connect to many destination computers.

Turcotte et al. in [251] models interaction counts between users and systems using the Poisson process. In particular, a Poisson factorization model for collaborative filtering is utilised and two user activities are considered: the processes run by the user, and machines on which users authenticate. A recommendation system is built for each of these observed activities and allow the model to capture different peer group structures to model each activity

type. Anomalies are reported using the Precision at N evaluation measure which is calculated for each user in the test set. In particular, the top N most anomalous users are reported and also a sensitivity analysis of the N parameter is performed. Their experimental results demonstrate better precision performance for recommending processes than authentications. A possible explanation is that the machines which users authenticate to are more sparse and diverse than the processes.

Price et al. in [203] present a change-point detection methodology to detect periodic sub-sequences. The authors claim that by separating sub-sequences that represent automated events from sub-sequences caused by human activity, the anomaly detection capabilities are enhanced. Change-point methods partition a sequence of data into smaller segments where each segment arises from a single generative model. Finally, the evaluation is performed on artificial data and real-world data; the LANL dataset [252] was used as real-world data. Their method is robust to duplicate and missing event data and identifies meaningful sub-sequences of event times.

1.3.3 Graph-based Anomaly Detection

Hagberg et al. in [99] represent authentication activity as a set of relationships between users and computers using graphs. As it was discussed in Sec. 1.1, LM is difficult to both detect and defend against and many studies introduce strategies to mitigate the risk associated with credential stealing. This study models authentication events as dynamic bipartite graphs in order to mitigate this risk. In particular, the authors compute the largest connected component of this graph as a quantitative measure of the network's vulnerability to such attacks. Their experiments show that an effective method to limit the number of credentials stored across networked computers was identified.

Amrouche et al. in [19] introduces a method that is appropriate for root cause analysis. They construct authentication graphs by using known malicious events. In particular, the authors investigate and visualize malicious authentication events and their proposed method could be used improve the existing solutions.

Kent et al. in [136] provide an analysis of how privileged and non-privileged users differ. They construct network authentication graphs and illustrate the difference in terms of complexity between a typical user without administrative access and a typical user with administrative access. For this illustration and for their analysis the LANL dataset was used. Finally they employed a logistic regression to determine inappropriate administrator-like behavior within the enterprise network. The AUC of this model is equal to 0.89.

Purvine et al. in [205] study cyber attacks which have a lateral movement component and propose a metric to recommend mitigation strategies to cyber analysts. In addition, a minimal

model that captures only the essential features of a generic lateral movement is introduced. It should me noted that, since networks are evolving over time the authors propose a dynamic threshold to be used to mitigate the lateral movement attack over time. All the experiments were performed on the LANL dataset and show the capabilities of their model.

1.4 Conclusion

In this chapter, we discussed the characteristics of the lateral movement attack and presented the techniques that intruders use to effectively execute such an attack. Moreover, existing works related to anomaly detection methods for the detection of the lateral movement attack were extensively discussed. More specifically, we discuss the main three categories of such anomaly detection works, namely machine learning-based, statistical-based and graph-based

.

Chapter 2

Supervised Learning Ensemble Method

Anomaly detection on security logs is receiving more and more attention. Authentication events are an important component of security logs, and being able to produce trustful and accurate predictions minimizes the effort of cyber-experts to stop false attacks. Authentication events could be classified as *Normal*, for legitimate user behavior, and *Malicious*, for malevolent behavior. These two behaviors (normal and malevolent) consistently produce imbalanced data which make the classification problem challenging.

In the commonly used real-world dataset for cyber-security research analysis, provided freely by the Los Alamos National Laboratory, the malicious behavior comprises only 0.00033% of the total. As such, the level of class skewness in this dataset creates a highly imbalanced scenario. This chapter addresses such a highly imbalanced scenario by introducing a novel feature engineering strategy followed by a ensemble supervised learning approach to further classify authentication events trustfully. The ensemble is composed of three uncorrelated classifiers (i) *Random Forest*, (ii) *LogitBoost* and (iii) *Logistic Regression*. Finally, the unweighted majority voting method is employed to leverage the individual predictions of the previous models to ultimately produce a final prediction for each authentication event. To the best of the authors knowledge, this chapter is the first attempt to address the supervised problem of detecting abnormal authentication events.

2.1 Introduction

The lateral movement attack is executed by repeatedly creating malicious authentication events. Malicious authentication events happen when attackers impersonate legitimate users by stealing their credentials, allowing them to acquire access to enterprise networks. Stealing credentials play a key role in cyber attacks and this is confirmed by Verizon's report [256] where 63% of confirmed data breaches involved leveraging weak/default/stolen credentials.

This chapter addresses the supervised learning problem of anomaly detection in order to accurately and trustfully classify authentication events. In particular, a novel method is introduced that aims at detecting individual malicious authentication events in order to defend against a possible lateral movement attack. A common characteristic of *authentication events* or *login logs* is being comprised of multidimensional categorical variables. Categorical variables stem from discrete entities and their properties, e.g. source user, destination computer, or protocol type. The underlying values of this type of variables are inherently unordered and as a consequence it is often hard to define similarity between different values of the same variable. Detecting anomalies on discrete data is challenging and is not a well studied topic[9] in the data mining field ; the primary focus is on continuous data.

One of the major challenges of detecting malicious authentication events (anomalous objects) is the inherited nature of of the problem. In particular, malicious authentication events are scarce relative to the events produced by normal network operations. As a consequence, the classes of the authentication events are highly imbalanced. A common approach to deal with class imbalance is the random under-sampling of the dominant class (the Normal class in our context), or the over-sampling of the under-represented class (the Malicious class in our context) by synthetically generating data observations. In this work, a random under-sample approach of the normal events was elected. Re-sampling techniques is a very active research topic where sophisticated techniques [188, 90] have potentiality of improving classification performance. Additionally, another challenge is to create new features, known as feature engineering, out of a purely categorical space.

The main research question guiding our efforts is: *"can we detect malicious authentications accurately and trustfully in a supervised learning setting?"*. Additionally, the main contributions of this work are the following:

- Advanced feature engineering using a graph-based model
- Fine grained classification of authentication events – as *Normal* or *Malicious* – instead of the common approach of classifying users
- A method for combining classification methods as an ensemble to predict authentication events trustfully

2.2 Overview of Methodology

The introduced supervised ensemble anomaly detector is composed of three supervised predictive algorithms *LogitBoost*, *Random Forest* and *Logistic Regression*. These supervised

algorithms are designed differently and by employing them we aim to capture different anomalous patterns. The final step of our methodology is to combine [151] the predictions of each classifier and the uniform weighted majority voting technique is used to combine these predictions. Fig. 2.1 illustrates the pipeline of our methodology.

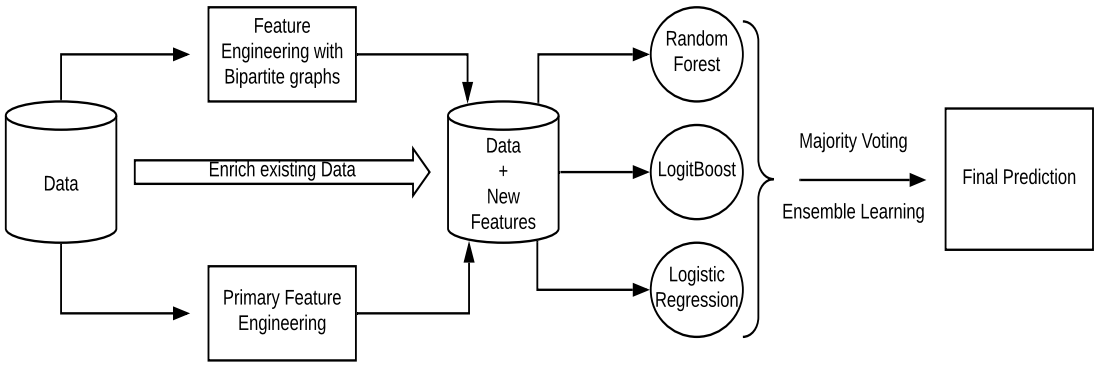

Fig. 2.1 Visual illustration of our methodology

The following sections provide details of the different steps of our method.

2.3 Primary Feature Engineering

In this section, we provide extensive in details regarding the advanced feature engineering of this work. The performance of a classification task is dependent on the ability of the features to reveal patterns that assist the classifiers in separating the classes. Additionally to the original features (columns of the given dataset) we extracted composite features, aiming to expose those patterns. To ease the comprehension of our feature engineering strategy, we formally define what an authentication event is:

Definition 1 ***Authentication Event****: An authentication event, e, is defined as a vector:*

$$e = < T, SrcUser, DstUsr, SrcCmptr, DstCmptr >$$

The individual elements are formally defined as:

- T: $time \in [0, time_{max}]$,
- $SrcUser@domain, DstUsr@domain \in U$,
- $SrcCmptr, DstCmptr \subset C$,
  $SrcCmptr \neq DstCmptr$

As a next step, we present three examples of an authentication event in a increasing T order:
e_1 = <50556, U1534, U1534, C13868, C1624>
e_2 = <50687, U1534, U1534, C13024, C1624>
e_3 = <50152, U832, U832, C3176, C2825>

Furthermore, we refer to the set of users and computers as C and U respectively. Additionally, we define a *Malicious User* as it follows:

Definition 2 ***Malicious User:*** *A user $u_i \in U$ is called a Malicious User if this user has produced at least 1 malicious authentication event in his entire user activity.*

The set of those features described below, is the result of extensive experimentation. We have identified the following necessary features and have decomposed them into tangible properties of the data.

- **Distribution of time difference of events between systems and from user to system**: captures the spread of activity over time
  - The *Median* of time differences in seconds.
  - The *95th Percentile* of time differences in seconds.
  - The *Standard Deviation* of time differences in seconds.
- **User activity and connection frequency**: describe the prevalence of network actions
  - The *Frequency* as the amount of past similar events
  - The *First Occurrence*, a flag denoting an event without any prior similar event
- **Distribution of Malicious events if we see every event as a trial**: quantify how probable is the first of success (Malicious event) when we observe a number of failures(Normal events).
  - The *Geometric Distribution* of malicious events within a sequence of similar events
- **User variance**: quantify the significance of a specific user within the history of events
  - The *Popular User* as the user value with the most occurrences within a sequence of similar events.

– The *Diversity* as the number of different users within a sequence of similar events.

We introduce the notion of similar events in order to calculate the aforementioned features. Similarity between two events is inspired by bipartite graphs and is a very important element of the introduced feature engineering design. Following we formally define what it means for two events to be similar.

Definition 3 ***Event similarity:*** *We define events e_i and e_j as similar when either of the following conditions are met:*

$$e_i \simeq e_j \Rightarrow \begin{cases} SrcUser_i = SrcUser_j \\ DstCmptr_i = DstCmptr_j \end{cases} \quad or \quad \begin{cases} SrcCmptr_i = SrcCmptr_j \\ DstCmptr_i = DstCmptr_j \end{cases} \tag{2.1}$$

Definition 2.1 is inspired by bipartite graphs and by following it all the aforementioned features can be calculated. The bipartite graphs are built using sets of events, per specific combinations of user and computer values. In particular, we refer to the set of all events in the dataset as E. Furthermore, we define the sets h_i and r_i, which are two supporting sets of every event e_i, as it follows:

Definition 4 ***Event history:*** *We define the history h_i of event e_i at time T_i as the set of events e_j at time T_j before T_i.*

$$\forall e_i, e_j \in E \rightarrow h_i = \{e_j : e_j \simeq e_i, 0 \leq T_j < T_i\}$$

Definition 5 ***Event recent past*** *We define the recent past r_i of event e_i at time T_i as the set of events e_j within one hour prior to T_i.*

$$\forall e_i, e_j \in E \rightarrow r_i = \{e_j : e_j \simeq e_i, 0 \leq T_i - 1h < T_j < T_i\}$$

It always holds that $r_i \subset h_i$, and $r_i = h_i$ when $T_i \leq 1h$.

2.4 Feature Engineering with bipartite graphs

For each event e_i in the dataset, we construct two tuples of bipartite graphs ($H_{U,i}$, $H_{C,i}$) and ($R_{U,i}$, $R_{C,i}$), constructed from h_i and r_i sets respectively. The new composite features will be

extracted from properties of those graphs. Each tuple consists of one graph ($H_{U,i}$ or $R_{U,i}$) using the features *SrcUser* and *DstCmptr* as left and right nodes respectively, and a second graph ($H_{C,i}$ or $R_{C,i}$) constructed in a similar manner using *SrcCmptr* and *DstCmptr* as nodes. To construct the nodes we use only those values that match those of the initial event e_i. An event involving two nodes on either graph constitutes an edge between them. Duplicate nodes and edges are merged but persist their information in an attribute A_i as a vector of tuples $A_i = [(Time_j, SrcUser_j)]$. A_i is constructed from the elements of all the events e_j that produce each graph. *SrcUser* is included in the A_i attribute of graphs $H_{U,i}$ and $R_{U,i}$ to maintain one algorithm for all graphs, albeit redundant.

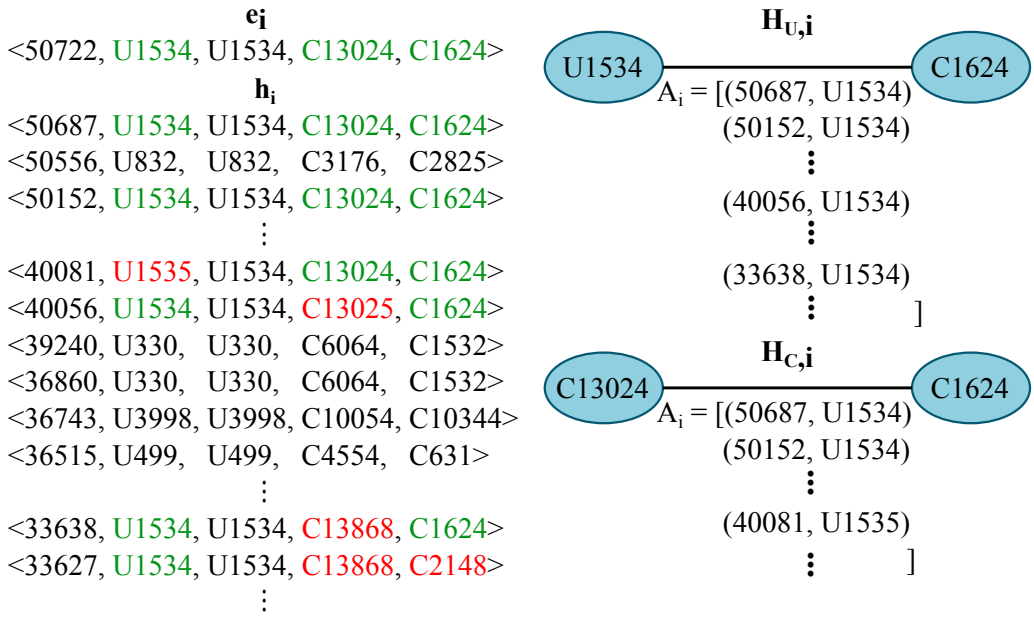

Fig. 2.2 Example of graphs $H_{U,i}$ and $H_{C,i}$ from an arbitrary excerpt of h_i, given e_i. Events that do not match the e_i values for the features used to build each graph are discarded.

Figure 2.2 presents a simple example of these graphs; notice that in graph $H_{U,i}$ of Figure 2.2, the event at time *40081* is filtered out, since the *SrcUser* feature does not match the initial event. As a consequence of its construction, each graph will only have 2 nodes and 1 edge.

In addition to the already calculated features, we use the A_i vectors to calculate the following mix of numeric and categorical features:

- The **Median**: the median of time differences.
- The **95th Percentile** of time differences in seconds.
- The **Standard Deviation** of time differences in seconds.

- The **Frequency** as the number of elements in vector A_i.
- The **First Occurrence** boolean denoting if A_i is empty.
- The **Geometric Distribution** where success is having a malicious event.

Using only graphs $H_{C,i}$ and $R_{C,i}$ we calculate the last two features:

- The **Popular User** as the user value with the most occurrences.
- The **Diversity** as the number of different users.

The calculated composite features provide the contextual modeling of the data which will enable our predictive models to enhance their accuracy. It should be emphasized that all the introduced composite features do not contain information for the future. It is very important to ensure that there is no information leakage.

2.5 Predictive Models

We use one algorithm from each sophisticated ensemble learning techniques so-called meta-algorithms: *Boosting* and *Bagging*. These techniques combine several machine learning algorithms into one predictive model in order to decrease the bias (boosting), and the variance (bagging). The bias is a part of the error caused by bad model and the variance is a part of the error caused by the data sample. Following are presented the selected classifiers:

- **LogitBoost** [87] belongs to the *Boosting* family of algorithms; it is based on decision trees, which are considered weak learners, but performs as a strong learner. It optimizes logistic loss instead of exponential loss.
- **Random Forest** [38] belongs to the *Bagging* machine learning algorithms, which reduce variance to avoid overfitting.
- **Logistic Regression** [140] measures the relationship between the categorical dependent variable and the independent variables by estimating probabilities using a logistic function.

The selected models have pairwise uncorrelated classification methodologies and produce uncorrelated predictions which are vital for building a good ensemble classifier. All the selected models are able to handle categorical variables without any encoding transformation. Finally, the majority voting [151, 195] method with uniform weights is applied in order to

combine the predictions. The majority voting technique classifies every event by extracting the most predicted class from the classifications of all other models. By combining uncorrelated predictions we not only build a good ensemble but we also increase the trust and robustness of our method. The increased trust in our proposed methodology arises from the fact that the reported classifications are achieved with the aid of three predictive models.

2.6 Training

The core ingredient that could affect the classification process is the skewness of the class distribution. During the training of each model, we randomly *under-sampled* the most prominent class and repeated 5 times the *10-fold Cross-validation* technique. *Under-sampling* the major class helps to train the models in a stratified way, by containing equal percentage of events of both classes, and to increase the performance of the classifiers on unbalanced datasets. Repeated *Cross-Validation* aims to avoid overfitting, making the models generalize well. Also, we tuned *LogitBoost* and *Random Forest* narrowly by finding the set of hyper-parameters that perform the best. For *LogitBoost* we tuned the number of boosting iterations (*nIter*) and for *Random Forest* we tuned the number of features that will be used to build the trees (*mtry*). *Logistic Regression* was applied without any parameter tuning.

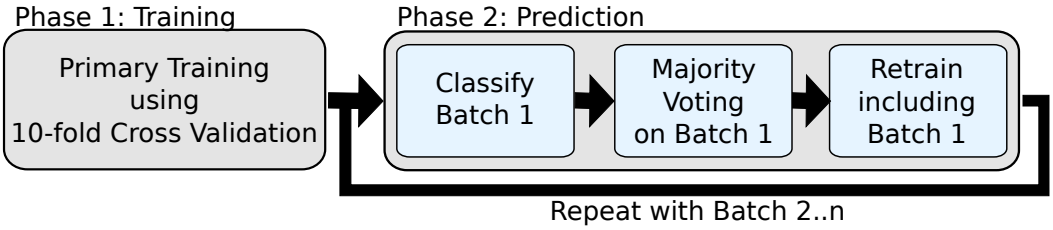

Fig. 2.3 Basic diagram of the training process

Multi-training updates the distribution of the primary dataset by extending the number of events. In detail it allows to:

1. take into consideration a real case of incoming new batch through the time,
2. measure the accuracy of prediction algorithms in the expanding dataset by predicting a fixed size dataset,
3. take into consideration the streaming data which is considered as a future work.

2.7 Conclusion

In this chapter, for the first time an ensemble supervised learning method is introduced to address the problem of detecting malicious authentication events. In particular, bipartite graphs were used to model the interactions between individual users and computers in order to perform feature engineering. Finally, three uncorrelated classifiers combined by following basic principles of ensemble learning to trustfully classify authentication events.

Chapter 3

Automatic Semi-supervised Ensemble Method

Cyber-attacks become more sophisticated and complex especially when adversaries steal user credentials to traverse the network of an organization.Detecting a breach is extremely difficult and this is confirmed by the findings of studies related to cyber-attacks on organizations. A study conducted last year by IBM found that it takes 206 days on average to US companies to detect a data breach. As a consequence, the effectiveness of existing defensive tools is in question.

In this chapter, we introduce an automatic semi-supervised ensemble method to detect malicious authentication events. The automatic nature of our methodology essentially springs from (i) the sequential procedure that is followed, (ii) the fact that the normal behavior is learnt by established unsupervised outlier ensemble theory. An one-class classification ensemble is developed by leveraging the knowledge of the normal behavior.

The main challenges that this chapter addresses are the following: 1. developing an effective outlier detection method on an excessively class imbalanced scenario, 2. developing an effective outlier detection method on a pure categorical feature space that is produced by the authentication event logs, 3. developing such a method that detected outliers are true malicious authentication events. The performance of our detector is evaluated on a real-world cyber security dataset provided publicly by the Los Alamos National Lab. In addition, by detecting malicious authentication events, compared to the majority of the existing works, which focus solely on detecting malicious users or computers, insights can be provided regarding when and at which systems malicious login events happened.

3.1 Introduction

The JP Morgan Chase [231] and Target hacks [144] are two well known examples of hacks where the adversaries stayed undetected while they traversed network. The lateral movement attack belongs to a category of attacks called **A**dvanced **P**ersistent **A**ttacks, where the prominent characteristic is that they are stealthy, well orchestrated and the adversaries stayed undetected for a long period of time. Specifically, during the execution of a lateral movement attack the adversaries gain shell access and make use of legitimate credentials to log into systems. Afterwards, they escalate privileges and subsequently manage to traverse a network without any detection.

Researchers have addressed the detection of malicious (unauthorized) authentication events by evaluating their methods on a real-world cyber security dataset provided freely by the Los Alamos National Lab [133]. Existing works focus on detecting malicious users or computers which leads to classifying all the generated events from a user or computer as malicious or legit. As a result, they fail to specify which events are malicious and to provide any information regarding at which systems the adversaries managed to impersonate benign users. Additionally, most of the existing approaches on this dataset are questionable and the authors in [204] provide further details.

A common characteristic of authentication events is being comprised of multidimensional categorical variables. Categorical variables stem from discrete entities and their properties, e.g. source user, destination computer, or protocol type. The underlying values of this type of variables are inherently unordered and as a consequence it is often challenging to define similarity between different values of the same variable. Moreover, the prominent challenge in the cyber defensive world is to develop effective approaches and sufficient labelled training sets are absent.

A possible solution to this point comes from the semi-supervised approaches [137] that do not require anomalous instances in the training phase. These approaches model the normal class and identify anomalies as the instances that diverge from the normal model. Instances only from the normal class (target class) are used during the learning phase in order to build the normal model. The task is to define a boundary around this class to minimize the chance of accepting objects from the anomaly class. Finally, the learnt model is used to assess if an unseen observation belongs to the target class or not.

In this chapter, the aim is to detect unauthorized events to services or computers in contrast to the majority of the existing work by analyzing freely available Los Alamos authentication dataset [133]. An embedding based and automatic semi-supervised outlier detector is introduced to reduce the false positives produced by an unsupervised outlier ensemble. In particular, our approach is comprised of two ensemble outlier detection

components that are connected in a sequential manner. An unsupervised outlier ensemble is developed to identify the most confident normal data points which afterwards feed an one-class classifier [242] to ultimately detect outliers. The authors in [13, 279] extensively discuss the details of ensemble learning for outlier detection tasks.

Additionally, the contributions of our proposed approach are:

- We produce an embedding space via the Logistic PCA [155] algorithm that has potentiality of better representing the normal behavior.
- We develop the Restricted Principal Bagging (*RPB*) technique, an improved variant of the well established feature bagging technique [159], that works on the principal components space.
- We introduce a new unsupervised combination function, *Vertical Horizontal Procedure* (*VHP*), that leverages gradually the predictions of individual and smaller scale ensemble members.
- We automatically build an automatic semi-supervised ensemble by combining the aforementioned novel components to effectively detect malicious events.

Overall, our approach improves current state-of-the-art methods and enhances the understanding related to the anomalous patterns produced by malicious authentication events. Detecting malicious events compared to previous works that detect malicious users or computers give us the opportunity to answer more actionable questions. The introduced method could also be used to extend existing methodologies, which detect malicious users or computers, to further detect individual malicious authentication events. To the best of our knowledge, this work is the first automatic semi-supervised attempt that aims at detecting anomalous authentication events.

3.2 Overview of Methodology

In this chapter, an automatic semi-supervised ensemble is introduced that is developed on categorical data produced by authentication events logs. In particular, the introduced method automatically creates the training set of an one-class classification ensemble. This training set is "non-polluted" by outliers and represents the normal behavior. More specifically, first it builds an unsupervised outlier ensemble to identify, with a relative confidence, authentication events that are normal. Secondly, it develops an one-class classification ensemble detector which learns a decision boundary around the normal class using only the predicted normal

authentication events derived from the first phase. Finally, the one-class classification ensemble classifies authentication events, that are not present in the training set, as belonging to the learned normal class or not. Figure 3.1 illustrates the sequential and automatic nature of our approach. Throughout this work we use *outliers* and *anomalies* interchangeably.

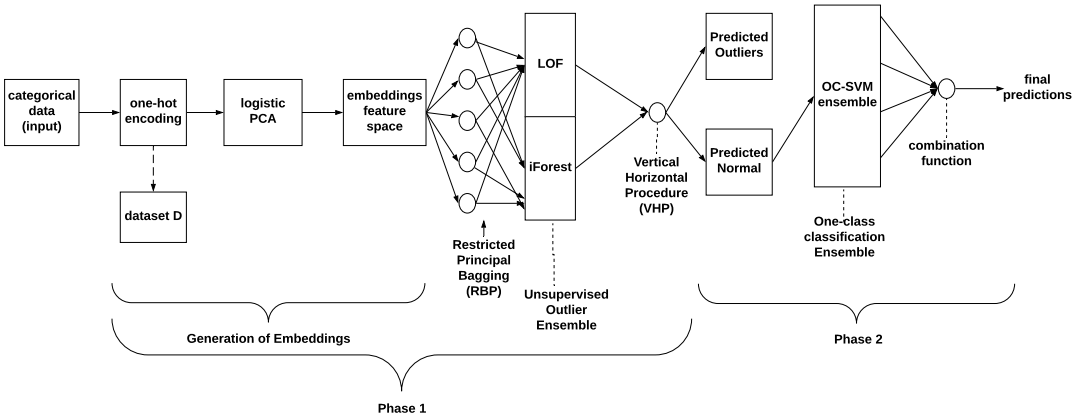

Fig. 3.1 Auto Semi-supervised Outlier Detector

3.3 Phase 1

Unsupervised outier detection algorithms detect outliers by scoring data according to their algorithmic design [282]. In this phase, we reverse the problem of leveraging outlier scores in order to identify the most outlier observations. The aim is to identify, with a relative confidence, non-outlier observations in order to be used as training sets by multiple one-class classifiers. In particular, the most non-outlier observations are identified by constructing an outlier ensemble on bagged subspaces and composed of two unsupervised scoring detectors.

3.3.1 Generation of Embeddings

Recently, word embeddings [185, 92, 163] have been introduced to map phrases from a vocabulary to vectors of real numbers. In this chapter, an embedding technique is followed to map objects from a categorical space with many dimensions to a continuous space with a much lower dimension. Specifically, authentication events produce a pure categorical space and the Logistic PCA algorithm [155] is applied to perform the mapping. As such, the produced principal components are leveraged in order to develop our outlier detection method. It should be noted, that before the Logistic PCA algorithm is applied, the categorical space has to be transformed into a feature space totally comprised of binary values.

A high percentage of explained variance by the principal components ensures that the embeddings space encloses information very close to the information included in the original variables. Also, we leave a sensitivity analysis related to number of principal components for the future. Additionally, according to Theorem 2 of [155] we select columns to decrease the deviance the most. In particular, the authors proved that: *For logistic PCA with* $k = 1$*, the standard basis vector which decreases deviance the most is the one corresponding to column with mean closest to* $1/2$. In addition, they proved that this theorem can be easily extended to k larger than 1 which is our case.

Furthermore, in Fig. 3.2 an example of mapping five data points from a binary feature space to a 2 dimensional continuous space using the Logistic PCA algorithm.

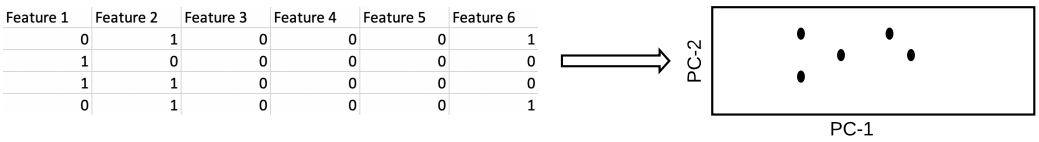

Fig. 3.2 An example of five data points embedded into to 2-D

3.3.2 Restricted Principal Bagging

Our motivation for developing the *RPB* (Restricted Principal Bagging) technique is to add randomness in a similar way like the Feature Bagging technique [159]; randomness is a key ingredient of outlier ensemble techniques [279]. Additionally, *RPB* is constructed in such way that upper bounds the sample space of the principal components. Moreover, our introduced technique aims at capturing the individual contribution of each principal component to the total explained variance. In other words, the Feature Bagging technique [159] is adjusted to work for principal components. Algorithm 1 presents in pseudo-code the steps of the *RPB* technique.

Firstly, we denote by *PCs* the principal components that we keep after we have applied the Theorem 2 of [155]. Afterwards, a set called V composed of all S_j is created, where $S_j = p * PCs$ and p is the percentage of the first p principal components. Afterwards, for each S_j and for $Iter$ iterations *RPB* samples from a uniform distribution U$(d/2, d-1)$ without replacement, where d is the dimensionality of S_j. Hence, for each $Iter$ iteration N_i principal components are randomly sampled and create a dataset F_i is created. Finally, an unsupervised outlier detector with random parameters is applied to F_i.

Algorithm 1 Restricted Principal Bagging

Input:

• **V** the set of all the S_j

• **OD** is an unsupervised Outlier Detection algorithm which outputs numeric outlier scores for each data point

• **Iter** represents how many times we perform feature sampling

Output

• **E** is a vector composed of oulier scores for each data point

Procedure:

1: **for** S_j in V **do**

2: **for** $i = 1,2,3,4,...Iter$ **do**

3: Randomly sample from a uniform distribution between $\lceil d/2 \rceil$ and $(d-1)$, where d is the number of the principal components in S_j

4: Randomly pick, without replacement, N_i principal components to create a subset F_i

5: Apply OD on F_i feature space

3.3.3 Unsupervised Outlier Detectors

Two well performing and established unsupervised detectors are combined to identify the most confident non-outlier (normal) observations. Our method intentionally selects heterogeneous detectors in order to capture different patterns of anomalies. It is worth noting that, our method remains identical in case more than two unsupervised detectors are selected to build the ensemble.

Firstly, the iForest [167] algorithm is employed which is a tree-based and state-of-the-art detector and performs the best across different contexts [78, 72, 264]. Secondly, the LOF [39] algorithm is employed which is a proximity-based method and designed to detect local outliers (see [13] for local and global outliers). It is a state-of-the-art outlier detection algorithm with an extensive body of research [159, 94, 18, 282].

It should be emphasized that our method is very flexible. As a consequence, *Phase 1* could be composed of a different number of unsupervised outlier detection algorithms. Also, the selected algorithms could be substituted by other unsupervised algorithms.

LOF and iForest independently apply the *RPB* technique on set V to build the ensemble version of LOF and iForest respectively. Henceforth, we call *LOF - RPB scores*$_j$ and *iForest - RPB scores*$_j$ the produced outlier scores by applying the *RPB* technique on a subset S_j and employing the LOF and iForest respectively. The final step is to combine these results and for this reason the *VHP* combination function is introduced.

3.3.4 VHP Combination Function

The *RPB* algorithm builds a couple of ensembles on each S_j called *LOF Ens* and *iForest Ens*. Hence, a combination function is introduced to effectively combine these ensembles instead of applying a global combination function across all the results; *LOF - RPB scores*$_j$ and *iForest - RPB scores*$_j$. The authors in [275] propose a novel local combination function and highlight its effectiveness compared to global functions.

In our strategy the average function is utilized to calculate the average scores of ensemble members. The average function is robust and widely used in the outlier ensemble literature [64]. Combining effectively outlier ensemble members without leveraging the ground truth is challenging and the authors in [13], [146], [279] extensively discuss the topic.

In particular, firstly all the *LOF - RPB scores*$_j$ and *iForest - RPB scores*$_j$ are normalized to z-scores before calculating the average scores of each S_j. As such for each subset S_j we build an ensemble produced by these combined *RPB* outlier scores. This ensemble is denoted by *LOF Ens & iForest Ens*. Afterwards, we convert the numeric outlier scores of each *LOF Ens & iForest Ens* ensemble (j in total) to binary values based on a threshold. Finally, we combine these binary values by utilizing the unweighted majority voting [254] technique to produce the output of *Phase 1*. Specifically, W is comprised of the most confident non-outlier (normal) data points and O the least confident non-outlier (normal) data points.

The conversion to binary values is referred as the *Vertical Strategy* and the combination of the binary values as the *Horizontal Strategy*. Henceforth, we call this combination function as *VHP*, Vertical Horizontal Procedure. All the outlier scores are normalized with the Z-score normalization scheme which is the most commonly used in outlier detection literature (see [13] for details in different normalization schemes).

3.4 Phase 2

This phase leverages the produced W dataset by *Phase 1* to build the semi-supervised ensemble. This dataset is composed of the most confident non-outlier (normal) data points which are used as training sets by multiple one-class classifiers. Hence, the introduced method is sequential and automatic at the same time. The desired outcome of *Phase 2* is to reduce significantly the number of false positives produced during *Phase 1*.

In particular, the One-Class SVM (OCSVM) algorithm [223] is used at *Phase 2*. This algorithm is one of the most widely used one-class classifiers and performs well on several problems [235, 158]. OCSVM is a boundary method that attempts to define a boundary around the training data (normal class), such that new observations that fall outside of this boundary are classified as outliers [236].

The training of the OCSVM algorithm is performed on the dataset W with multiple and independent executions with random parameter values. Afterwards, each training execution the algorithm is tested on dataset O and an outlier score vector is produced. This vector has length equal to the number of observations of O dataset. Finally, all the produced outlier score vectors are combined to ultimately produce the final outlier score vector. The procedure of running a detector over a range of parameters and combining the produced results is interpreted as an ensemble(see [13] for details). Overall, both *Phase 1* and *Phase 2* of our proposed approach are developed without leveraging the ground truth.

3.5 Conclusion

In this chapter, an automatic semi-supervised detector for malicious authentication detection was introduced that outperformed existent supervised approaches and tools with the human in the loop. The proposed method was able to capture all the underlying mechanisms that produced anomalous authentication events. Moreover, it managed not to miss any malicious authentication event by evaluating it on the most widely used real-world authentication log dataset. In addition, it demonstrated improvement compared to state-of-the-art methods. Also, the conducted sensitivity analysis of *Phase 1* showed that the user defined threshold of the ranked outliers did not affect at all the true positive rate of *Phase 1*. On the other hand, regarding the true positive rate there is some effect which is not so noticeable.

Chapter 4

Experiments, Results and Discussion

In this chapter, we elaborate on a widely used and freely available real-world dataset related to authentication events. This dataset enabled us to validate our research methods for the purposes of cyber security. In addition, the experimental setting and experimental results of the ensemble learning detectors, introduced in Chapters 2 and 3, are explained in detail. The objective of the conducted experiments is to demonstrate the effectiveness of our introduced ensembles in a cyber security application related to detecting malicious authentication events. Furthermore, not only we provide comparisons of the introduced ensembles of this thesis with the state-of-the-art but also we compare them against each other. Finally, this chapter concludes with insights gained from our analysis.

4.1 Dataset

This section provides details related to the dataset we used to address the problem of detecting malicious authentication events. Specifically, the Los Alamos National Laboratory (LANL) provides a publicly accessible comprehensive dataset [134, 133] to be used for cyber security analytical purposes[1]. Its content was collected over a period of 58 consecutive days and is comprised of 1.05 billion authentication events (70GB is the total uncompressed size) from multiple sources, such as individual computers, servers, and Active Directory servers running the Microsoft Windows operating system. Following are presented the attributes of the authentication events:

- **Time:** Timestamp of the event
- **Source User@Domain:** Which specific user is launching the event.

[1] It is available at *https://csr.lanl.gov/data/cyber1/*

- **Destination User@Domain:** The user that the event is terminating at.
- **Source Computer:** Which specific computer is originating the event.
- **Destination Computer:** The computer that the event is terminating at.
- **Authentication Type:** Authentication events can have several types such as Negotiate, Kerberos and NTLM based on authentication protocol type.
- **Logon Type:** The logon type of event could be an Interactive keyboard session, a Batch event, a system Service, a screen saver Lock or Unlock and several others. Missing values might occur because of the undetermined logon type.
- **Authentication Orientation:** This attribute indicates whether it is a Kerberos TGT or TGS, a log on or log off event.
- **Success or Failure:** This attribute indicates whether the authentication event was successful or not.

In addition, the authors in [134, 133] provide a data element that presents bad behaviour; the *RedTeam* table. A group of authorised attackers, commonly known as a *red team* is responsible for creating the events in this table. In particular, 749 authentication events are known to have been performed by the red team using stolen user credentials. This table could be used as ground truth to classify the authentication events as malicious or normal. In Fig. 4.1 the counts of user and computer is illustrated. There is a strong periodicity where non-work days show lower counts. On the other hand, the total volume of the authentication events remains consistent.

4.2 Hardware & Software

The experiments were performed in the R and Python programming languages. Also, a 2.4 GHz Intel Xeon E5, 50 GB RAM, running Ubuntu 16.04 machine was used to carry out all the experiments.
Supervised Learning Ensemble Method: The modeling and the *feature engineering* implemented by using the packages *caret package* [261] and *data.table package* [73] respectively.
One-Class Classification Ensemble Method: The logisticPCA [155] R package was used for the implementation of the Logistic PCA algorithm and the data.table [74] R package for fast data manipulation. The iForest, LOF and OCSVM algorithms were executed using the Python scikit-learn library [199].

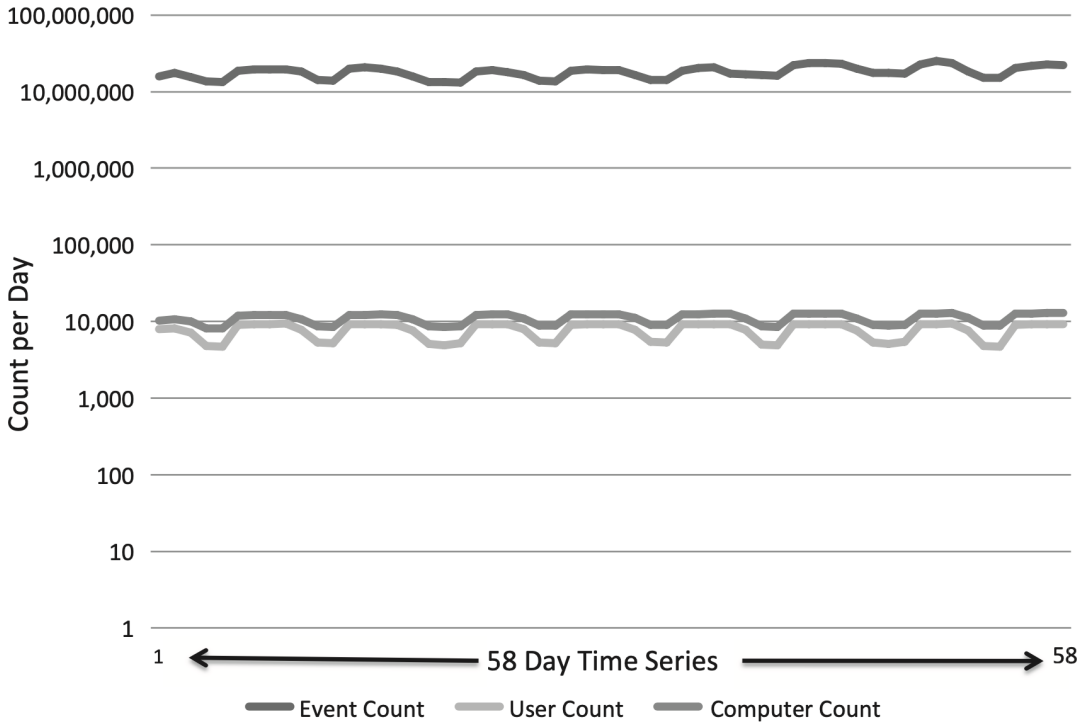

Fig. 4.1 Authentication volume by computer, user and event count per day (58 days in total). [135]

4.3 Supervised Learning Ensemble Method

In this section, we present the experimental results of our approach introduced in Chapter 2.

4.3.1 Data

As presented in 4.1, the dimensionality of the complete dataset is exceptionally large and as a consequence it is challenging to be processed without adequate big data infrastructure. As such, sampling was performed to overcome this challenge and introduce our method. In particular, 21 out of the 98 malicious users were randomly selected in order to obtain adequate malicious authentication events. It is worth noting that, even the malicious activity of malicious users is excessively scarce. This is confirmed by the following finding: The corresponding percentage of each malicious user in our random sample is less than 0.65%. Furthermore, Table 4.1 presents the level of class imbalance for those randomly selected users by categorizing the absolute number of malicious events.

In addition to the aforementioned sampling, we filter out the authentication events that are *Local*. Authentication events are characterized as *Local* or *Remote* if the source and destination computer values are the same or different respectively. *Local* compared to *Remote*

Table 4.1 User event class comparison

Malicious Events	avg % of total p.u.
1	0.035
(1,9]	0.119
(9,19]	0.291
(19,inf)	0.245

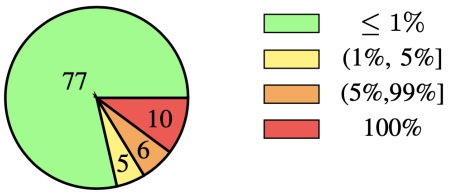

Fig. 4.2 Number of users (out of 98) grouped by their percentage of malicious events over their total events

events are potentially less harmful. This is conformed by our analysis that showed that all malicious events of our sample are of *Remote* type.

We analyzed the 30 first days of authentication events and the corresponding percentage of the malicious class is equal to 0.00033% out of the total number of events. We disaggregated the total skewness of the class distribution to the skewness for all the malicious users. Figure 4.2 shows the relative amount of source users grouped by their percentage of malicious events over their total events. The overwhelming majority has less than 1% of malicious activity within the 30 day subset.

4.3.2 Evaluation

In our experiments, we have followed the multi-training procedure (see subsection 2.6 for details) where at each iteration the prediction dataset size is extended by 700 new events. In addition, the training dataset contains 12.5 consecutive days or 199090 authentication events. The prediction set is composed of 37 iterations, with a events batch size of 700 each, or 25900 total events. The events in each batch are consecutive over time. The evaluation of our methodology is performed using the following metrics and measures. The positive class is composed of *Malicious* events.

- **False Positive Rate** (FPR): Normal events misclassified as Malicious.
- **False Negative Rate** (FNR): Malicious events misclassified as Normal.

- **Balanced Accuracy** (BACC): the arithmetic mean of True Positive Rate (TPR) and True Negative Rate (TNR) – also known as "Strength" [46]. It is a different way to measure correct classification rate.
- **Positive Predictive Value** (PPV): $\frac{\text{Number of True Positives}}{\text{Number of Predicted Positives}}$. It represents the probability that a person has a disease or condition given a positive test result.
- **F1-measure** (F1): the harmonic mean of TPR and PPV.
- **Prevalence** (Prev): the ratio of Positive condition size (TP + FN) over the sample size [46].

For the purposes of our analysis, we group separately the iterations which introduce events of both classes (Group A) and those with events from only the Normal class (Group B). Most of the aforementioned metrics require both classes for their application, allowing in-depth evaluation of the classifier performance. Group B iterations can be evaluated only via FPR, however having Group B iterations i) models true network conditions more realistically, and ii) emphasizes the significance of the achieved FPR and FNR in Group A. We first analyze Group A, which consists of 7 out of the 37 total iterations. Afterwards, we go through Group B; in the absence of true malicious events, only the FPR and Prev metrics can be evaluated.

4.3.3 Results and Comparative Analysis

Results

Table 4.2 presents the 99% bootstrap confidence interval and standard deviation of the average false positive rate of each individual classifier and the produced ensemble classifier.

Table 4.2 Bootstrap CI 99%

Models	**Lower**	**Upper**	**Std Dev**
Random Forest	0.0016	0.0141	0.0027
LogitBoost	0.0007	0.0069	0.0014
Logistic Regression	*0.0002*	*0.0022*	*0.0005*
Ensemble	0.0004	0.0032	0.0007

Both classes - Group A: Majority Voting produced a 0 *FNR*; in other words, no malicious events were misclassified. Class imbalance has a considerable effect on these cases, hence the perfect *FNR* is a noteworthy result.

Table 4.3 presents the performance metrics considering the Majority Voting results. The average *FPR* is 0.0076 and the standard deviation of the rate is 0.00942. The *BACC* metric

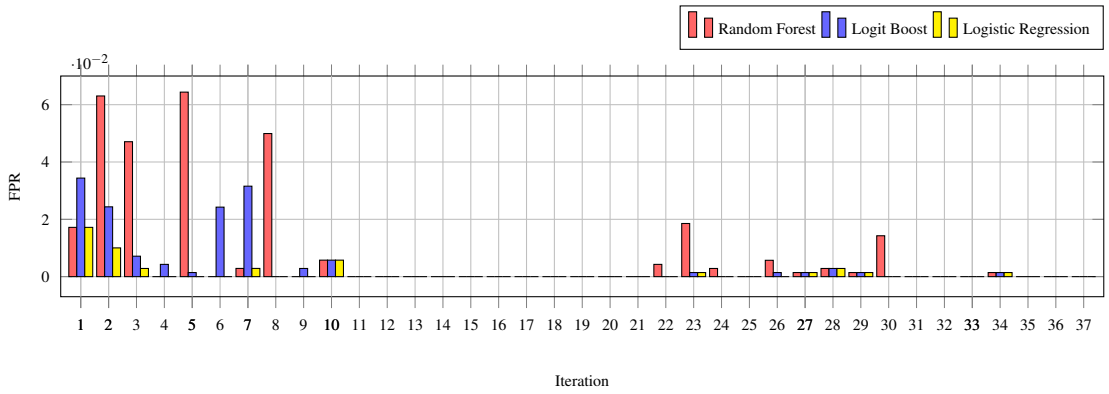

Fig. 4.3 Bar chart of the false positive rate per model over all iterations. Group A Iterations are in bold

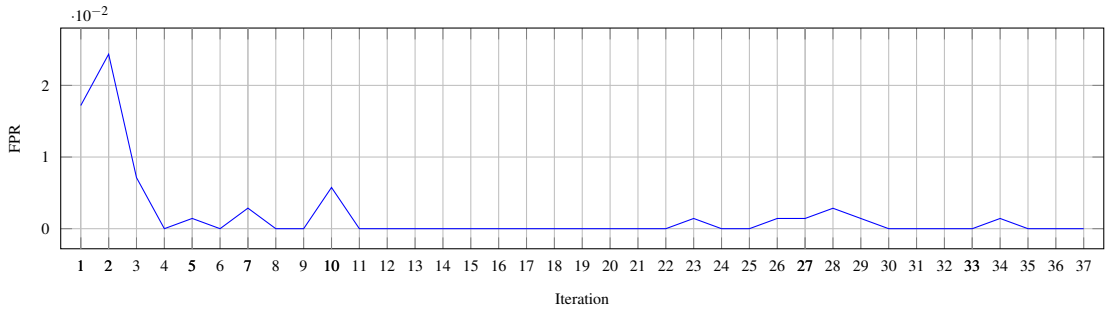

Fig. 4.4 False Positive Rate produced by the ensemble classifier for all iterations. Group A Iterations are in bold

is on average 99.62%. *F1* is on average 0.6587. *Prev* of *Malicious* events is very low and affects the predictive values. For instance, if *Prev* doubles, while the *TPR* and the *PPV* stay the same, the *F1* increases by 16%.

Only normal class - Group B: Majority Voting produced 0 *FPR* for 24 out of 30 prediction iterations; no Normal event was misclassified as malicious. The average *FPR* of all the iterations is 0.0026 and *Prev* is constantly 0. Figure 4.4 presents the Majority Voting FPR for each iteration. The aggregated average FPR is 0.0019. The FPR per classifier is shown in Figure 4.3.

Comparative Analysis

Our method introduced in Chapter 2 was the first ensemble supervised learning approach with graph-based feature engineering that aims to detect lateral movement attacks. Recently, Zhenyu Bai in [26] employed several machine learning algorithms (supervised and unsupervised) using the major part of feature engineering work. The exact list of the features

Table 4.3 Performance Metrics for the prediction iterations of Group A

Iteration	**FPR**	**BACC**	**PPV**	**Prev**
1	0.0172	0.9914	0.2	0.0043
2	0.0243	0.9878	0.15	0.0043
5	0.0014	0.9993	0.66	0.0028
7	0.0029	0.9986	0.66	0.0057
10	0.0057	0.9971	0.6	0.0086
27	0.0014	0.9993	0.5	0.0014
33	**0.0000**	**1.000**	**1.0**	**0.0014**

Table 4.4 Our supervised learning ensemble vs Zhenyu Bai [26]. *: Model validation without user-name, source and destination features

Classifier	**Accuracy**	**Precision**	**Recall**	F_1	**Training Time (s)**
Zhenyu Bai	**99.99%**	99.87%	**99.73%**	**0.998**	11.28
Our Ensemble	99.98%	**100%**	98.67%	0.993	20.48
*Zhenyu Bai	**99.99%**	99.87	99.47%	0.992	**10.53**
*Our Ensemble	99.98%	**100%**	90.66%	0.951	18.19

that was used is provided in [26]. Zhenyu Bai implemented our approach and compared his LogitBoost [87] classifier against our ensemble. Table 4.4 presents the performance of our ensemble and Zhenyu Bai's method and the best results are in bold. The recall and F_1 scores of our model are slightly lower than Zhenyu Bai's but precision is better. The training time of our ensemble, which is composed of three individual models, is close to double the training times of Zhenyu Bai's method. Furthermore, the author in [26] exclude few attributes of the events logs (see Sec. 4.1 for details) and presents the corresponding results. There is a significant drop in terms of performance for our ensemble when these attributes are excluded. However, excluding attributes from the modelling phase without performing feature selection is not an advisable approach.

Bian et al. in [31] implemented the semi-supervised learning detector of Chen et al. in [61] and our supervised learning detector to further evaluate their method. Table 4.5 presents the performance of the three competitors where the best results are in bold. In particular, our ensemble perrforms better than Bian et al. in precision and marginally improves the F_1 measure. However, our feature engineering and model training times are magnitudes higher than both Chen et al. and Bian et al.. In addition, Bian et al. evaluate the robustness of the aforementioned approaches on never seen data. Table 4.6 shows that the model of Bian et al. model outperforms the competitors and report better generalization. However, their implemented version of our ensemble does not include the re-sampling

Table 4.5 Our supervised learning ensemble vs Bian et al. [31] and Chen et al. [61]

Classifier	Precision	Recall	F_1	Feature Extraction Time (s)	Training Time (s)
Bian et al.	97.02%	93.04%	0.95	169.35	**1.45**
Chen et al.	73.12%	7.24%	0.13	**0.69**	5.29
Our Ensemble	**100%**	**93.47%**	**0.97**	100.81	23332.37

Table 4.6 Our supervised learning ensemble vs Bian et al. [31] and Chen et al. [61]

Classifier	Precision	Recall	F_1	Feature Extraction Time (s)	Training Time (s)
Bian et al.	**61.24%**	**94.05%**	**0.74**	475.46	2.56
Chen et al.	4.64%	9.52%	0.06	**11.22**	**0.66**
Our Ensemble	9.58%	45.83%	0.16	40488.56	1903.24

technique that we utilized. Hence, this might have affected the performance of our model because in highly imbalanced data it is crucial to balance the classes.

4.3.4 Discussion

Group A iterations showcase the extreme imbalance of the classes in the data with a prevalence of 0.0040 on average. However, our method manages to correctly classify all malicious events. From a cyber-security perspective this translates into a 100% successful detection of true attacks. Unfortunately, the 0.0019 FPR translates into 49.2 Normal events falsely classified as Malicious for the subset we analyzed.

If we accept the 0.0019 FPR value as a potential constant result, it would give 1995000 falsely classified events over the total dataset of 58 days, or 34396 events per day, an 81% reduction over the total average events per day from the Los Alamos dataset. This number is huge for any security team to handle manually, and the absolute number of events is too large to block indiscriminately. Nonetheless, being able to reduce the search space by 81% is a great improvement for any cyber-security expert trying to detect attacks in a busy network. In addition, the harmonic average probability of a Malicious classified event to be truly Malicious is 0.6587, indicating that our approach performs above average and is a promising first step.

An important factor to consider when using supervised algorithms is the frequency for updating the model.

We have selected 700 events as our iteration length which corresponds to on average 54 minutes of consecutive events. The motivation of selecting 700 events was to find an appropriate number of events that maximally spreads Malicious events over different iterations. It should be noted that, it was an arbitrarily picked number that gave a sufficient

spread and wasn't the result of thorough experimentation. Furthermore, network intrusion attacks are usually spread so thinly over time that iterations lacking true Malicious events is highly probable regardless of iteration length. An analysis over the effect of the iteration length is worth exploring and will be included in subsequent publications.

4.4 Automatic Semi-supervised Ensemble Method

In this section, we present the experimental results of our approach introduced in Chapter 3.

4.4.1 Data

As presented in 4.1, the dimensionality of the complete dataset is exceptionally large and as a result it is challenging to be processed without adequate big data infrastructure. Additionally, except for the set of attributes that the LANL dataset in 4.1 is comprised of, a new attribute is created based on the case where the source computer and destination computer are the same or different. This new boolean feature quantifies the Local or Remote rule respectively. In addition, the time variable is excluded from the conducted analysis and as a result a purely categorical feature space is produced.

Sampling from an excessively imbalanced dataset usually produces samples composed of observations belonging solely to the prominent class. As a consequence, it is impossible to evaluate the performance of method on detecting anomalous objects. Additionally, an extensive experimentation regarding the scalability of the selected unsupervised and one-class classification algorithms is conducted. A sample composed of $150,000$ consecutive authentication events is considered a good choice. Also, this sample has to contain at least 5 malicious events in order to thoroughly evaluate the anomalous class. As such, the resulting data sample is composed of 150,000 consecutive authentication events and the total number of the malicious events is equal to 10. As such, the corresponding percentage of malicious events is 0.0066%. Finally, the one-hot technique is applied to produce the input space of the Logistic PCA algorithm. The dimensionality of the resulting binary space is $150,000 \times 2700$ and we refer to this dataset as D.

4.4.2 Experimental Setting

The major objective of our experiments is to investigate the effectiveness of our proposed auto semi-supervised detector compared to state-of-the-art works. Additionally, the conducted experiments do not leverage the ground truth to tune the performance of our detector.

Generation of Embeddings

The dimensionality of dataset D, that is the output of the one-hot procedure, is $150,000 \times 2700$. The Logistic PCA is applied on D and 900 principal components that explain 93% of the total variance, are kept. Afterwards, we apply Theorem 2 that explained in Sec. 3.3.1 to reduce the number of the principal components. Finally, the total number of principal components is 500 and those will be the embeddings feature space. Henceforth, we refer to the embeddings feature space as *PCs*.

Phase 1

As extensively discussed in 3.3, this phase involves the utilization of two unsupervised outlier detection algorithms; LOF [39] and iForest [167]. LOF is an algorithm that is neighborhood based and as a consequence the neighborhood size is the only input parameter. iForest is a tree based algorithm based on recursive partitioning and its input parameters that can be adjusted are more than one. In particular, the exact parameter values of each algorithm are presented in Table 4.7. LOF is employed with different neighborhood size and iForest is employed with the following cartesian product $IF = \{(Number\ Of\ Estimators \times Maximum\ Samples \times Maximum\ Features)\}$.

It should be noted that, the outcome of this phase is an ensemble which constructed in an unsupervised manner. Hence, no ground truth is leveraged to find the best performing parameters. In an unsupervised and ensemble setting, the algorithms run multiple times with random parameters. Afterwards, their predictions are combined in an unsupervised way in order to construct the ensemble. Hence, the challenge now is how to effectively combine individual predictions. The fundamental principles to build a good unsupervised ensemble are presented in [282, 279, 12, 282].

Table 4.7 Setting parameters

	subsets S	**Parameters**
LOF	V = {4% ,10% ,20%, 30%, 40%, 100%}	$Neighbors$ = {5,10,15,20,30,40,50,60,70,80,90,100}
iForest	V = {4%, 10%, 20%, 30%, 40%, 100%}	$NumberEstimators$ = {100, 200, 300, 400} $MaximumFeatures$ = {10%, 20%, 40%, 60%} $MaximumSamples$ = {10%, 30%, 50%}

Finally, we refer to the ensemble of *Phase 1* as ***VHP-Ensemble***. This ensemble, uses the *RPB* algorithm to create bagged spaces and afterwards combines the results using the introduced *VHP* function. Also, we refer to the baseline model as the ***Vanilla-Ensemble***. This ensemble leverages the whole *PCs* embeddings space and uses the feature bagging technique introduced by Lazarevic [159]. The combination function is the average function. Table 4.8 presents in detail the components of both ensembles.

Table 4.8 Ensembles of *Phase 1*

	Detector		principal components of subsets S							combination		Bagging	
Ensmbles	**LOF**	**iForest**	**20**	**50**	**100**	**150**	**150**	**200**	**500**	**VHP**	**Avg.**	**RPB**	**Lazarevic**
VHP	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No	**Yes**	No	**Yes**	No
Vanilla	**Yes**	**Yes**	No	No	No	No	No	No	**Yes**	No	**Yes**	No	**Yes**

Table 4.9 Setting parameters

nu	{0.0001, 0.0005, 0.001, 0.005}
gamma	{0.01, 0.05, 0.09, 0.001}
kernel	{"rbf", "sigmoid"}

Phase 2

As extensively discussed in Sec. 3.4, this phase involves the utilization of the OCSVM algorithm. In the same analogy as in *Phase 1* the one-class classification ensemble is developed in an unsupervised manner. Table 4.9 presents in detail the parameter values that the OCSVM algorithm uses. The authors of the OCSVM algorithm [223] conclude that $\nu \in (0,1]$ and γ are the most suitable parameters to tune. Additionally, except for the ν and γ parameters we select different kernel functions. The ν parameter represents an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors and the γ parameter represents the kernel coefficient. Hence, in our experiments we use the gaussian and the sigmoid kernel and we optimize the ν and γ parameters. Finally, the average function is utilized to combine the outlier scores of all OCSVM executions.

4.4.3 Evaluation

Recall at N (R@N) and Precision at N (P@N), are widely used for evaluation in outlier detection [13, 45] and are appropriate for binary predictions. In particular, Precision at n is defined in [66] as the proportion of correct results in the top n ranks. For a dataset D of size N, consisting of outliers O $\subset$ D and inliers I $\subseteq$ D (D = O $\cup$ I), P@n can be formalized as:n

$$P@n = \frac{|o \in O | rank(o) \leq n|}{n} \tag{4.1}$$

On the other hand, Recall at N (R@N) computes the fraction of ground-truth positives, which are captured in the top n ranks. Between recall and precision there is clear trade-off which means that the better the precision the worse the recall and vice versa.

Table 4.10 Precision and Recall of the output of *Phase 1*

Enembles	$P_{@1500}$	$R_{@1500}$	$P_{@5000}$	$R_{@5000}$	$P_{@7000}$	$R_{@7000}$
VHP	**0.015**	**1.0**	**0.008**	**1.0**	**0.007**	**1.0**
Vanilla	0.005	0.8	0.0016	0.8	0.0011	0.8

4.4.4 Results, Comparative Analysis and Discussion

Results of Phase 1

Table 4.10 summarizes the performance of the *VHP-Ensemble* and *Vanilla-Ensemble*. The output of *Phase 1* is two sets, W and O which represent the most and least confident outlier objects respectively. Hence, the precision and recall evaluation measures are appropriate measures.

In addition, the threshold of the top n ranks is adjusted to showcase the sensitivity of our introduced ensemble. *VHP-Ensemble* is developed in a pure unsupervised setup so it is important to investigate the sensitivity of the n parameter in top n ranks of outliers. As such, we investigate the effect of *Phase 1* on building the semi-supervised ensemble detector. Hence, different numbers of top n ranks of outliers in P@N and R@N are reported. In our analysis N plays the role of the relative confidence to identify the non-outlier data points. Table 4.10 presents the performance of the *VHP-Ensemble* with three different numbers of top n ranks of outliers (top 1500, top 5000 and top 7000 data points are considered as outliers) and the *Vanilla-Ensemble*. The performance of the ensembles is a typical example of the trade off between precision and recall. In our proposed approach the cost of higher precision is less than the cost of higher recall.

Results of Phase 2 and Comparison

The performance of the ensemble constructed at *Phase 2*, which represents the performance of our introduced methodology, is demonstrated and compared with existing works. As such, previous works that focus on detecting malicious authentication events are considered as competitors. Unfortunately, existing works that focus on detecting malicious users or computers can not be considered as competitors because a huge amount of events have to be further analyzed to identify which specific events are malicious. In addition since the number of existing works on malicious events is limited, we decided to enlarge the competitors list by comparing our proposed detector with any machine learning scenario (supervised, semi-supervised, unsupervised) that is tested on authentication events. Hence, we evaluate our method with i) Siadati et al. [230], ii) Lopez et al. [172], iii) our supervised learning work [128] presented in Chap. 2

We denote by *Detector-1500* the semi-supervised ensemble detector, which is developed when a threshold rank N=1500 is chosen for the *VHP-Ensemble*. The most outlier point among the N=1500 reported outliers has a rank of 1. In the same fashion, we develop *Detector-5000* and *Detector-7000* where N=5000 and N=7000 respectively. Our motivation for selecting so large N is that we want to feed the semi-supervised detector with the most confident normal data points. We identify them based on our intuition for the outliers percentage in our dataset. In our case, N is at least 150 times greater than the number of true malicious authentication events.

In Fig. 4.5 a summary of the FPR and TPR scores produced by all the competitors is presented. Amongst the competitors, Siadati et al. [230] achieves the lowest FPR whereas our supervised learning work [128] and all the variants of our semi-supervised method achieve the highest TPR; they do not miss any malicious login. In addition, *Detector-1500* achieves the lowest FPR among all the competitors. Ultimately, *Detector-1500* improves the FPR of our supervised detector by 10% (150 login events) and more than doubles Siadati's TPR. Siadati et al. detector is based on integrating security analysts knowledge into the detection system in the form of rules that define login patterns. In other words, this detector does not improve the existing knowledge of the cyber analysts for anomalous patterns, but instead relies on known rules to detect anomalies. As a consequence, the Siadati's rule based visualization detector misses 53% of the malicious logins.

In addition, each of the aforementioned approaches outperforms the logisitic classifier of Lopez et al. [172] which achieves AUC 82.79%. We do not plot their reported FPR and TPR scores in figure 4.5 because their FPR scores are at least 5 times worse than the maximum FPR value in figure 4.5. Consequently, we avoid presenting a figure that is less readable and informative for the majority of the competitors.

4.5 Conclusion

In this chapter, two novel ensemble methods were evaluated to address the problem of malicious authentication events detection. In literature, there is a limited work that targets the prediction of malicious authentication events with the aid of machine learning algorithms.

First, a supervised learning ensemble anomaly detection method was proposed. Authentication events were modeled as bipartite graphs in order extract knowledge with feature engineering techniques. Overall, the proposed ensemble achieved a zero false negative rate and a zero false positive rate for 68% of the prediction steps. In addition, on average low values of false positive rate achieved for the remaining 32% prediction steps. The excessive class imbalance of the authentication event makes false positives an unavoidable problem.

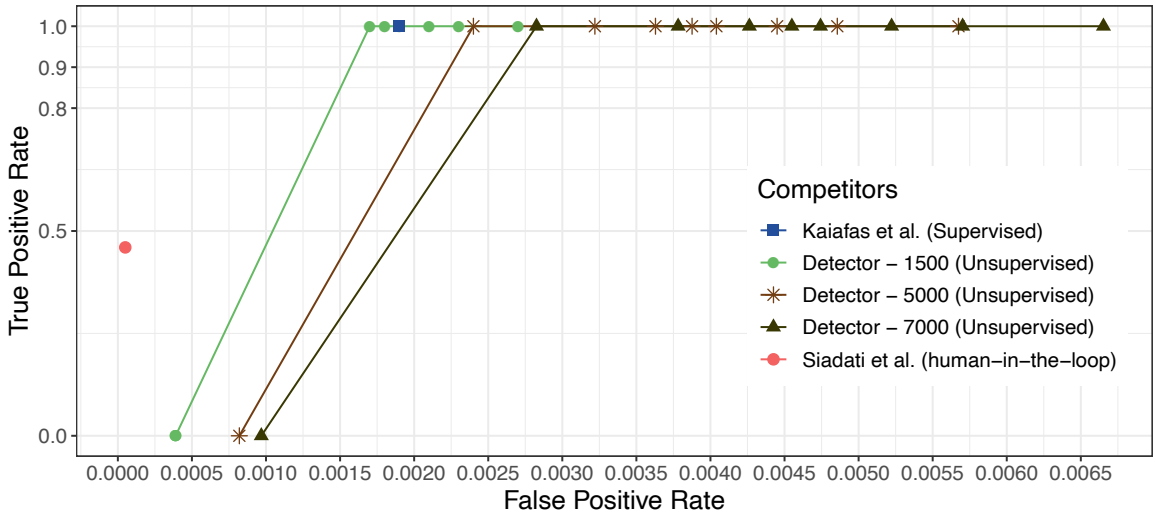

Fig. 4.5 Comparison of the Auto Semi-supervised Outlier Detector

On the other hand, the proposed method managed to control the false positive rate in low levels. Hence, this method managed to control the error rate in low levels and classify each event in a trustful way.

Second, an automatic semi-supervised ensemble anomaly detection method was proposed. This method managed to outperform the state-of-the-art that is related to detecting malicious authentication using supervised learning algorithms and tools with the human in the loop. The ensemble was tested on a real-world authentication events dataset and demonstrated that it did not miss any malicious login event. In addition, the one-class classification ensemble of *Phase 2* improved the false positive rate of the unsupervised ensemble of *Phase 1* almost 9 times. Also, the conducted sensitivity analysis showed that the rank threshold at *Phase 1* did not affect at the true positive rate of the one-class classification ensemble at *Phase 2*. In particular, all the ensembles that we constructed based on different levels of rank thresholds did not miss any true malicious login events; true positive rate is equal to 1. On the other hand, the false positive rate did not affected noticeably by the rank threshold.

Part II

Detection of Telecommunication Fraud

Chapter 1

Background and Related Work

In this chapter, we elaborate on the fraud ecosystem and discuss the different fraud areas i.e. telecommunication fraud, bank fraud etc. In addition, we present the challenges and issues that fraud detection systems have to deal with. It is of paramount importance to know in advance the most common issues that a fraud detection system has to overcome to be as effective as possible. Furthermore, we focus on telephony fraud and provide details regarding the most used types of telephony fraud schemes. Finally, this chapter presents existing works for detecting telephony fraud with data mining techniques. It should be noted that, there is little academic work in telecommunication area and this is confirmed by multiple studies.

1.1 Fraud Ecosystem

Fraud could be described as an intentional deception or misrepresentation that results in some unauthorized benefit to the fraudster or another person [115]. As such, any technological system that involves money and services can potentially be fraudsters' target, i.e. the credit card system and telecommunication system [17]. Recently, Abdallah et al. in their survey paper [4] presented statistics of published works, from 1994 to 2014, related to the most prominent fraud areas. Fig. 1.1 illustrates these statistics and shows that the least studied area, between 1994 and 2014, is the telecommunication area. The fact that there is little academic work in telecommunication area is also confirmed by Sahin et al. in [216] and Sahin's thesis [82]. Following we present the reasons for lacking academic work in the telecommunication area based on their findings:

- The complexity of the telecommunication network
- The lack of publicly accessible data for conducting experiments

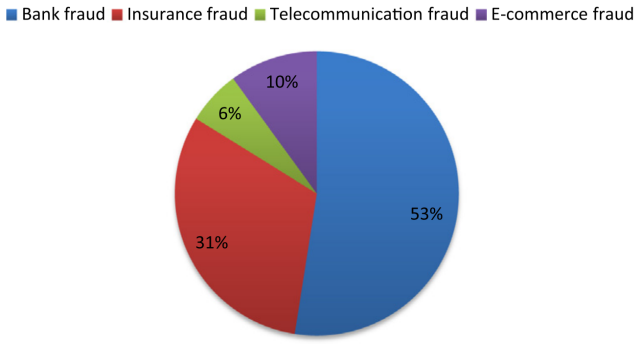

Fig. 1.1 Overview for the quantity of most researched area of fraud [4]

- The privacy constraints

In addition, across all fraud areas two mechanisms are in common and their goal is to defend against fraud activities. More specifically, **fraud detection** and **fraud prevention** are two layers of defence that can be found in every fraud area. As such, it is important to provide details in order to distinguish between these two layers. Fraud prevention describes measures to avoid fraud occurrence and secures the technological systems against fraud. In particular, it is the first defensive layer that aims to restrict, suppress, destruct, destroy, control, remove, or prevent the occurrence of fraud. An example of a fraud prevention mechanism is encryption algorithms that are applied to communication data. Fraud detection systems are the next layer of protection and can only be applied once a fraud activity has occurred. In other words, fraud detection systems have an effect once fraud prevention has failed. Moreover, fraud detection could be informally defined as the process of identifying fraud activities as quickly as possible. Therefore, effective fraud detection systems are being developed by integrating data mining methods in order to surpass the limited capabilities of systems that heavily depend on predefined a and subjective rules stated by experts [164]. Finally, a survey paper from Phua et al. [201] categorises, compares, and summarises data mining-based fraud detection methods and techniques published between 2000 and 2010.

In particular, anomaly or outlier detection methods are a sub-group of data mining methods interested in finding interesting data objects deviating in their behavior considerably from the majority and, as such, providing new insights. As such, fraud detection systems take advantage of such methods to identify any deviation from the norm and ultimately detect fraud. Developing a fraud detection system based on data mining methods is challenging and finding the most appropriate approach is subject to multiple factors. The most important factor that affects the selection of the approach is at which extend we are aware of the fraud activities. Hence, anomaly detection methods can be categorized into three groups based on the existing fraud knowledge; unsupervised, semi-supervised and supervised (see chapter 2

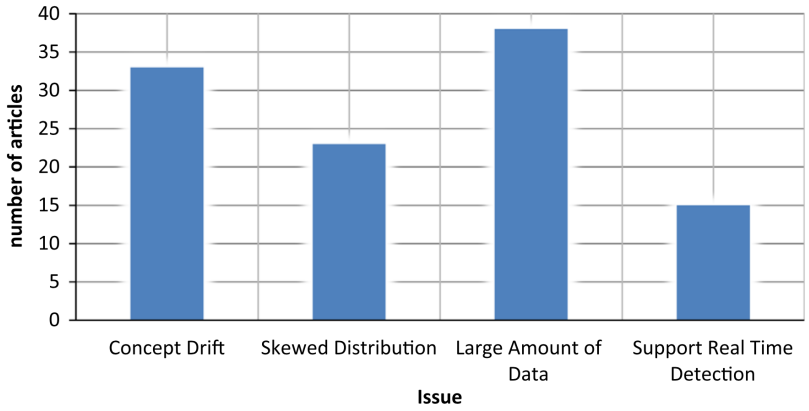

Fig. 1.2 Distribution of fraud detection articles based on issues and challenges [4]

for details). Abdallah et al. in their survey paper [4] summarize the multiple challenges that fraud detection methods have to deal with in order to be robust enough. Fig. 1.2 illustrates the distribution of published fraud detection articles based on issues and challenges. The concept drift [260, 249, 89] and the large amount of data are the most frequent issues that fraud detection systems face.

1.2 Telephony Fraud

A survey conducted last year found that compared to 2017, fraud losses as a percent of global telecommunication revenues grew 37% to $28.3 Billion USD, or 1.74% of total revenues. In addition, this study found that almost half of fraud departments are less than 7 years old which leads us to the conclusion that more and more organizations are interested in detecting fraud activities. In this section, we elaborate on telephony fraud, we describe the most used types of telephony fraud schemes.

Fraud in the telecommunication industry comes in many different forms with subscription fraud being the biggest concern for telecommunication operators. More specifically, subscription fraud, as it alone amounted to a loss of 5.22 billion dollars in the US in 2013 [23]. Subscription fraud occurs when a fraudster uses stolen identity credentials or provides fake information to obtain mobile services with no intention to pay. The problem of detecting subscription fraud has been addressed using data mining methods [220, 81] and classification methods based on privacy-preserving [110].

In PBX dial-through fraud, compromised PBXs can be used to make free calls, while the call charges are attributed to the PBX owner. **P**rivate **B**ranch **E**xchange system allows enterprise customers to manage their internal and external communication needs. A PBX is made up of both hardware and software that connects to communication devices such

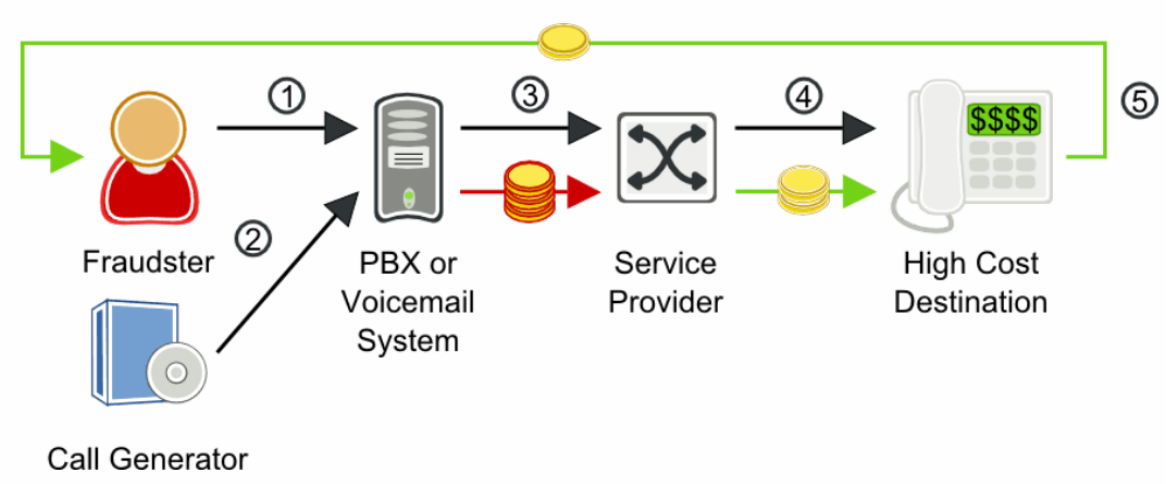

Fig. 1.3 Fraudsters are hacking an enterprise PBX to forward calls to a high cost destination

as telephone adapters, hubs, switches, routers, and telephone sets. IP-PBX systems can be compromised by malware or accessing an IP address connected with the PBX box to bypass the company's firewalls. In general, IP-PBX systems are vulnerable to the same threats as those that affect any data network including, DoS attacks and interception of communications. Fig. 1.3 illustrates a scenario where fraudsters hack a PBX to place calls to a high cost destination. Several works address the problem of toll evasion fraud against IP-PBX or demonstrate IP-PBX vulnerabilities [98, 269, 181, 182, 268]. The main difference between PBX and IP-PBX is the way the provide connection. A PBX uses standard telephone lines whereas an IP-PBX uses the Internet Protocol (IP).

Over-The-Top (OTT) services (e.g., WhatsApp, Viber, Zoom) use the internet to implement services without involving any telecommunication operators. OTT bypass or OTT hijack is a fraud technique where a normal phone call is diverted over IP to a voice chat application on a mobile phone, instead of being terminated over the normal telecommunication infrastructure[215]. More specifically, the OTT provider have to partner with a transit operator to hijack regular calls [82]. This rerouting (or hijack) is performed without explicit authorization from the all the involved parties; caller, callee and their operators. As such, the fraudsters manage to collect a large share of the call charge and induce a significant loss of revenue to the bypassed operators. Ighneiwa et al. in [117] developed an machine learning based method to detect bypass fraud.

International Revenue Share Fraud (IRSF), is one of the most problematic types of fraud [82]. In IRSF, calls to certain destinations are hijacked by fraudulent operators and diverted to the so-called 'international premium rate services'. Premium Rate Numbers (PRN) are used to provide wide range of services such as gambling, live chat; through voice

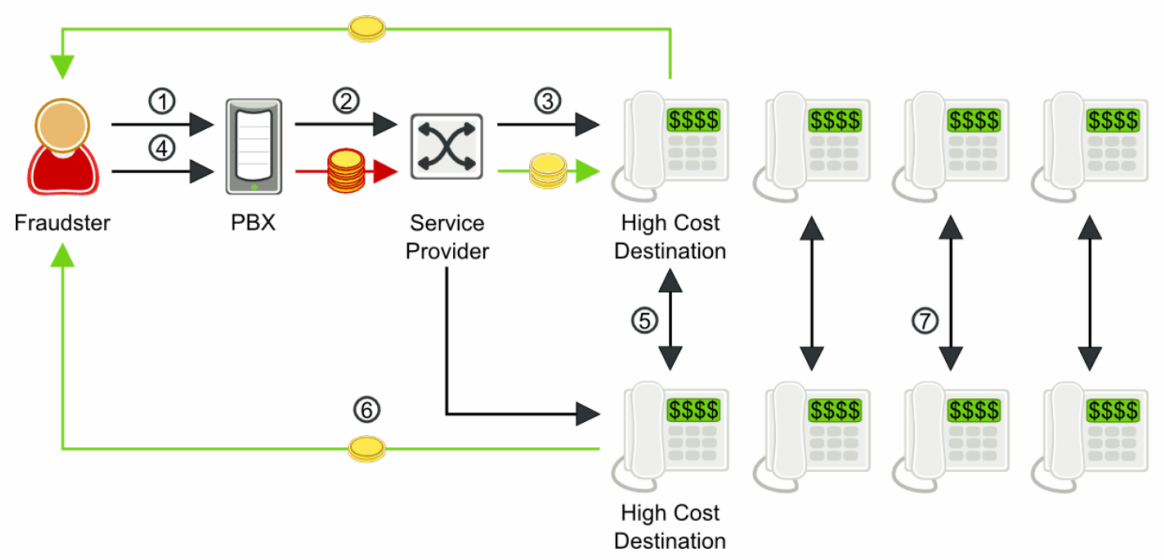

Fig. 1.4 Multiple call transfer fraud scenario

call or SMS. The cost of calling a premium rate number is much higher than a regular call to cover the cost of services provided [82]. Finally, after hijacking the fraudster generates high traffic calls to high cost destinations and gets revenue from the sharing agreements. Fig. 1.4 illustrates a multiple call transfers scenario. The attacker hacks an enterprise PBX to set up hundreds of simultaneous calls to high cost destinations.

Voice spam is one of the most visible types of voice fraud targeting customers and consist significant annoyances for telephone users. Disseminating telephone spam co-evolves with technology and as a result disseminating telephone spam has never been easier. As such, *robocalling* automatically dials and delivers a prerecorded message to a list of phone numbers [250]. Voice spam can take many forms, and recently Badawi et al. in [25] provide the first systematic study related to the Game Hack Scam (GHS) where the scammer promise unlimited resources or other advantages for their favorite game. In general, an effective execution of spamming (regardless of the medium) is composed of three basic elements: a recipient list, content, and a mass distribution channel [250]. Several works have been developed to address the problem of voice spam [265, 180, 118]

1.3 Related Work

In this section, existing works related to anomaly detection with data mining methods for fraud detection of telecommunication will be discussed. In particular, due to the fact that (i) telephony fraud takes many different forms, (ii) there is little academic work related to

telecommunication fraud, we present existing works developed using data mining techniques. However, due to the aforementioned reasons, we are not able to focus on any specific type of telecommunication fraud.

Taniguchi et al. in [239] present supervised and unsupervised approaches to detect fraud. First, a feed-forward neural network (supervised learning) is used to classify subscribers as fraudulent and legitimate. Secondly, a Gaussian mixture model (unsupervised learning) is applied to model past behavior of each subscriber and abnormalities are detected based on the past model. Lastly, two Bayesian networks (unsupervised learning) are used to model the behavior of fraudulent and legitimate subscribers. Their models demonstrate at least 0.7 true positive rate while the the false positive rate is zero. They dataset stems from call records used for billing purposes but it is described poorly and it is private. Additionally, each of the aforementioned approaches is applied on different set of features which leave us with the impression that the analysis is biased.

Hilas et al. in [111] investigate the effectiveness of supervised and unsupervised learning approaches to the problem of fraud detection in the telecommunications area. More specifically, feed-forward neural network (supervised learning) was used to classify users as normal and fraudster, and hierarchical agglomerative clustering (unsupervised learning) to test whether cases from the same class tend to form coherent clusters. Before applying their models the principal component analysis was performed to reduce the dimensionality and produce uncorrelated feature vectors. Their methods are evaluated on a real-world data (private) that covers a period of eight years and is composed of call detail records (CDRs). CDRs (see [112] for details) contain information such as: the caller ID, the chargeable duration of the call, the called party ID, the date and the time of the call etc. A user's data are aggregated in different ways to construct five profiles for each user and find the appropriate user profile (behavior characterization). Overall, the supervised approach managed achieve 0.8 true positive rates and 0.02 false positive rate. They also highlight the lack of interpretability of the feed-forward neural network models.

Elmi et al. in [77] develop a feed forward neural network (supervised learning) to detect a specific type of telecommunication fraud, namely the SIM box fraud. A SIM box is VoIP device that maps the call from VoIP to a SIM card of the same mobile operator of the destination mobile. The aim of their work is to detect SIM box fraud subscribers. In particular, they alter the architecture of the neural network to investigate the learning potentials of neural networks and identify useful features. The dataset is a CDR based dataset (private) but the authors do not provide information regarding its origin and how it was obtained. In addition, there is lack of transparency regarding the procedure followed to identify the final set of

features that effectively detects SIM box fraud activity. The experimental results demonstrate high classification accuracy but they fail to provide insights.

Furthermore, Elmi et al. in [219] extended their previous work [77] in detecting SIM box fraud. In particular, except for the artificial neural network (ANN) algorithm they also employ the SVM algorithm to conduct a comparative analysis. The experimental results show that SVM outperforms ANN in terms of accuracy, false positive rate and training duration.

Farvaresh et al. in [81] propose a sequential multi-phased method that aims at detecting subscription fraud on a real-world scenario. In the first phase data cleaning and the PCA algorithm are applied. Afterwards, the K-means algorithm is used to cluster the data where the optimal number of clusters and initial centers are found with the SOM (self-organizing map) algorithm. Moreover, clustering results are used as inputs to the last phase; a supervised learning approach. Specifically, the feature space is augmented with three new features. At the third and final phase several classification algorithms and approaches are developed. In particular, the decision tree, SVM, neural network, random forest and boosting trees algorithms are applied. Also, ensemble approaches such as stacking and majority voting are implemented. Finally, the authors justify the need of the multiple phases by comparing the performance of the classifiers with and without the corresponding algorithms at each phase. For instance, they compare the performance of models with and without using clustering features. Their experiments are performed on a dataset (private) composed of CDR and financial information coming from the Telecommunication Company of Iran. Based on their experimental results, the boosting tree algorithm is the best performing model which achieves 0.948 AUC.

Xing et al. in [266] propose a generative statistical model called LDA (Latent Dirichlet Allocation) to build user profile signatures of the normal behaviour. As such, fraudsters are detected via detecting deviations from normal behaviour. The LDA algorithm answers questions related to if a call is generated by a specific account given the feature values of the call. The basic idea of the LDA is to represent accounts as random mixtures over latent classes, where a latent class is characterized by a multinomial distribution. The data used for their experiments consisted of 67 days of CDR for the city of Glasgow. Their proposed method outperforms Taniguchi's method [239] and reports a rejection rate at the level of 2.25%.

Furthermore, Papadimitriou et al. [194] extend the work of Xing et al. [266] to detect fraudulent behavior. More specifically, they use the LDA algorithm and introduce four methods for approximating the KL-divergence between two LDAs. In addition, they compare their method with Taniguchi's gaussian mixture model [239] and show improvement. Their methods achieve 0.9833 AUC whereas Taniguchi's 0.9111.

1.4 Conclusion

Several works have been developed using data mining techniques but a significant part of them require a considerable amount of fraud knowledge. Hence, the practicality of these works is in question due to lack of such knowledge in fraud detection problems. In addition, it is apparent that the telecommunication area has not fully taken advantage of the advancements in the machine learning field to develop sophisticated fraud detection techniques.

In the next chapter, we address the problem of fraud detection in the telecommunication area by constructing unsupervised outlier ensembles. More specifically, we follow well-established theory of unsupervised outlier ensembles [279, 10] to conduct an experimental and comparative analysis. In an unsupervised setting there is no need for fraud knowledge to develop a method. To the best our knowledge, this analysis is the first that addresses the problem of the telecommunication fraud detection with outlier detection algorithms. As such, we differentiate ourselves from all the existing works.

Chapter 2

Unsupervised Ensemble Learning

Today, cellular networks and Voice over IP (VoIP) technology are incorporated into the global telephony network and provide many different services. In particular, **Private Branch Exchange** (**PBX**) is a technology that enables enterprise customers to manage their internal and external communication needs. This technology, as well as many other technologies, could be vulnerable to fraud activities in order to gain financial benefits. However, valuable telecommunication data sources are of paramount importance to develop effective fraud detection methods. More specifically, every time a call is placed on a telecommunications network, descriptive information about the call is saved. This descriptive information is called Call Detail Records (CDR) and is related to each call routed (originated, terminated or transited) over the network of an operator. CDRs include various information, such as originating and destination phone numbers, inbound and outbound routes, date, call duration and call type.

In this chapter, we deal with the detection of fraudulent private branch exchange phone calls made on the network of the largest provider in Luxembourg, POST Luxembourg. Established unsupervised learning principles are followed to address the challenging problem of fraud detection. In particular, an experimental research is conducted to investigate the performance of unsupervised outlier detection algorithms in a real-world fraud detection problem. For use in real-world business applications it is important to obtain a robust detection method, i.e. a method that can perform well on different types of data, to ensure that the method will not impact that business in unexpected ways. As such, unsupervised outlier ensembles are developed to expose the factors that affect the robustness of an outlier detection approach. Overall, our analysis demonstrates that despite the collective power of outlier ensembles they are still affected by i) data normalization schemes, ii) combination functions iii) outlier detection algorithms.

2.1 Introduction

Outlier detection is the process of identifying those observations that deviate substantially from the remaining data. In particular, identifying outliers in high-dimensional data can provide important insights into many real-world applications, e.g., detection of frauds, sensor failures, or outlying gene expressions. Choosing an unsupervised outlier detection method over a supervised or semi-supervised approaches is heavily influenced by the availability of labels [94].

Supervised outlier detection methods need sufficient labeled training and test sets. However, in real-world outlier detection problems the ground-truth related to the outlier class is missing. Outliers are naturally scarce and as a result such data will be heavily imbalanced which poses a problem for most classifiers. Most importantly, supervised methods struggle to detect novel types of anomalies because no labeled training examples have been collected.

Unsupervised outlier detection approaches do not rely on labeled datasets. Each outlier detection algorithm is based on a model making specific assumptions on the nature of outliers. Hence, every algorithm is able to capture specific data patterns and fits only to some aspects of the total ground truth. In other words, the subjectivity of each model influences the outlier detection performance.

Outlier ensembles take advantage of the individual subjectivity of each algorithm by combining several different outlier detection results to build more robust detectors. The authors in [13, 279] point out the challenges related to developing good outlier ensembles. The two major challenges are:

- *How to deal with accuracy?* Since in an unsupervised learning setting no ground truth is available the evaluation is difficult.
- *How to assess diversity?* Since in an unsupervised learning setting no ground truth is available diversity is usually based on randomness. The diversity of ensemble members is one of the most important ingredients of good ensembles [279].

Telecommunication frauds are malicious usage and/or exploitation of telephone connections for criminal purposes.These can range from finance gain for fraudsters to damaging public reputation of enterprises. More often than not, they cause substantial financial loses for victims. In this chapter, we deal with the detection of fraudulent private phone exchange (PBX) phone calls made on the network of the largest provider in Luxembourg, POST Luxembourg. In particular, we focus on unauthorized calls to international premium-rate numbers. These calls can cause large costs within short time frames. Quick and reliable detection and mitigation of fraudulent calls is therefore extremely important. We use Call

Detail Records (CDRs) logs to develop unsupervised outlier ensembles in order to identify fraudulent calls. Unsupervised learning approaches compared to supervised learning methods are not able to find the best performing parameters in order to tune their performance. The authors in [13, 8, 283, 279] present the core elements of good outlier ensembles and following these elements are summarized:

- **Data normalization** is followed to scale each attribute to [0,1] or to make it follow a $N(0,1)$. The authors in [45] show that outlier detection algorithms perform better on normalized datasets in contrast to unnormalized datasets.
- **Subspace outlier detection** to avoid irrelevant attributes and learn diverse models
- **Normalization of outlier scores** to make comparable the scores produced by heterogeneous outlier detection algorithms
- **Combination functions** to combine the outlier scores of the ensemble members.

The motivation of our analysis is to experimentally investigate the effect of the aforementioned core elements in the detection of fraudulent calls. Hence, (i) four different transformations are used to perform **data normalization**, (ii) the feature bagging technique [159] is used as the **subspace outlier detection** technique, (iii) the Z-score method is used to perform **normalization of outlier scores**, (iv) the average and maximum **combination functions** are used to combine the outlier score vectors.

The analysis conducted here helps us identifying selection criteria for robust ensemble unsupervised methods. Ultimately, this analysis is an important and mandatory step towards the automation of hybrid supervised learning approaches guided by outlier ensembles without the involvement of domain experts in the modelling phase.

2.2 Methodology

In this section we are giving details about the methodological part of our analysis and we are guided by a primary question: "What is the performance effect on an unsupervised outlier ensemble when it is constructed with different variants of the same ensemble core elements?".

Developing an unsupervised outlier ensemble is challenging when it comes to decide what will be the best performing alternative components. In this section bagging outlier ensembles are constructed with all possible combinations of the four core elements discussed in Sec. 2.1.

2.2.1 Data normalization

One of the main pre-processing steps for many statistical learning tasks is data normalization. The authors in [45] show that outlier detection methods on normalized datasets perform better compared to non normalized datasets. We normalize all the numerical attributes based on the following schemes.

- **Minimum and maximum normalization (Min-Max)** transforms the numerical attributes x of a dataset based on the formula: $\frac{x-min(x)}{max(x)-min(x)}$. This transformation is linear and each numerical attribute is mapped to the [0,1] range.
- **Mean and standard deviation normalization (Mean-SD)** transforms the numerical attributes x of a dataset based on the formula: $\frac{x-mean(x)}{sd(x)}$. This transformation is also known as the Z-score transformation and ensures that the attributes distributions have mean = 0 and std = 1.
- **Median and the IQR normalization (Median-IQR)** transforms the numerical attributes x of a dataset based on the formula: $\frac{x-median(x)}{IQR(x)}$ where $IQR(x)$ is the interquartile range of x. This transformation maps the attributes to have median = 0 and std = 1.
- **Median and median absolute deviation normalization (Median-MAD)** transforms the numerical attributes x of a dataset based on the formula: $\frac{x-median(x)}{MAD(x)}$ where $MAD(x) = median(\mid x - median(x) \mid)$ and each x is transformed to have median = 0 and std = 1.

As such, in our analysis the effect of four normalization schemes is investigated in relation with the detection performance of unsupervised outlier ensembles.

2.2.2 Subspace Outlier Detection

In our analysis, the feature bagging [159] technique is used as a subspace technique to discover relevant subspaces. In feature bagging, an outlier detection algorithm is applied to various random lower dimensional projections, i.e. using only a subset of the available features. As such, at each projected space the outlier detection algorithm scores the data according to their exceptionality. Afterwards, the outlier scores from the projected spaces are combined to produce a final outlier score vector; the target vector. In the rest of this chapter we refer to an outlier detection algorithm as *detector* or *base detector*.

2.2.3 Normalization of Outlier Scores

Different detectors may often score data on different numeric scales. Therefore, before combining the outlier scores of heterogeneous detectors the normalization of outlier scores is mandatory. Otherwise, some algorithms might dominate in the combination score. In addition, the ordering obtained by the outlier scores should be the same for all the detectors. Hence, inversion of the outlier scores might be needed.

In our analysis, the Z-score normalization scheme is employed in order to normalize the outlier scores of all detectors. The motivation of using this specific normalization is that the authors in [13] suggest that using Z-scores turns out to be quite effective in many settings. It should be noted that, in cases where a detector produces smaller scores as indicators of greater outlierness the negative of the Z-value should be used.

2.2.4 Combination functions

Ensemble learning methods combine the predictions from different base detectors in order to create more robust results. As such, ensemble model is often more powerful than the individual detectors. In our analysis, we use combination functions to unify the outlier scores obtained by the feature bagging technique. In particular, a detector with different parameter values is employed on random and lower dimension projections of the data. Afterwards, the produced outlier scores are combined to ultimately develop the corresponding unsupervised outlier ensemble.

In our analysis, the *mean of scores* and the *maximum of scores* are used as combination functions. The authors in [13] explain the benefits of using the mean and maximum as combinations functions. As such, the effect of two widely used combination functions is investigated in relation with the detection performance.

2.2.5 Assessing Diversity

The benefit of diverse outlier scores is that the true result will be close to detectors' predicted truth if they are accurate to some extent. Hence, our strategy is to select such detectors that the produced outlier scores obtain great diversity.

The Our work takes into consideration the findings of [224, 279] in order to select the detectors that generate dissimilar scores. In addition, we increase the diversity of the models by inducing randomness to each detector. More specifically, random parameters are used for each detector with the feature bagging technique. Moreover, based on the analysis of [45] we select detectors that their detection biases are different. For instance, if it was

to select the KNN [206] algorithm we should have avoided to also select the KNNW [21] algorithm. The outlier scores of these algorithms are highly correlated. Finally, the feature bagging technique produces uncorrelated results by inducing randomness on finding lower data projections [281].

2.2.6 Detectors

The strategy for selecting the detectors and perform our experimental analysis discussed on 2.2.5 section. In principle, however, one could choose any detector to perform a similar analysis as long as the strategy is the same. Hence, the following algorithms are the base detectors of the conducted experimental analysis.

1. **KDEOS** (Kernel Density Estimation Outlier Score) [226], computes a kernel density estimation over a user-given range of k-nearest neighbors. In particular, the gaussian kernel is used to estimate the density.

2. **LoOP** (Local Outlier Probabilities) [145], computes a local density based on probabilistic set distance for observations, with one parameter the k-nearest neighbors. The density is compared to the density of the respective nearest neighbors, resulting in the local outlier probability.

3. **iForest** (Isolation Forest) [167] detects anomalies in a tree ensemble fashion. It isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

2.2.7 Pipeline of constructing Bagging Ensembles

In this section we are describing the steps followed to construct the outlier ensembles of experimental analysis. First, we normalize all the numerical attributes based on the four formulas we discussed in 2.2.1. Afterwards, the feature bagging technique is applied as the subspace outlier detection technique. Next, the produced the outlier scores are normalized employing the Z-score function. Finally, the average and maximum combination functions are used to unify the outlier scores and ultimately construct the outlier ensembles. Therefore, given a detector and all the possible combinations of the aforementioned components, eight outlier ensembles can be constructed. As such, our experimental analysis is comprised of 24 ensembles are constructed which stem from the three detectors, KDEOS, LoOP, iForest. Fig. 2.1 illustrates the procedure that is followed to construct the ensembles of our analysis. In blue is coloured the process that is followed to construct one particular ensemble. Henceforth, we refer to each of the 24 ensembles as *Bagging Ensemble*.

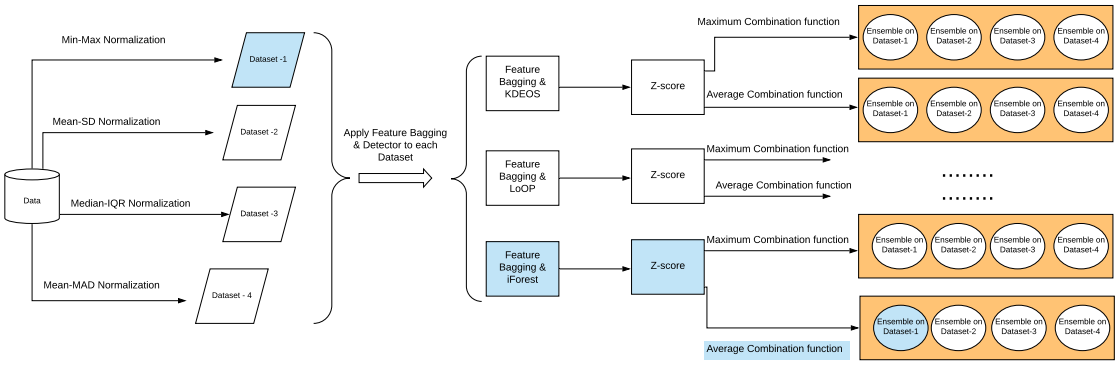

Fig. 2.1 The Pipeline of constructing *Bagging Ensembles*

2.3 Experimental Setup

2.3.1 Dataset Description

Our analysis uses a dataset comprised of CDRs (Call Detail Records) related to private branch exchange phone calls made on the network of POST Luxembourg. This dataset is a high-dimensional dataset which includes information related to calling number, calling time, calling duration, number of called parties, total calling cost and destination countries of the called parties. Henceforth, we refer to this dataset as D. CDR data are mainly used for billing purposes, but in this research such data are used for fraud detection and mitigation purposes. It is worth noting that, due to GDPR compliance rules POST Luxembourg has applied aggregation by time windows of 10 minutes and anonymization before the experimental analysis. Additionally, Table 2.1 presents details regarding the fields and notation of D.

Table 2.1 Notation

Field	Notation
Average number of distinct calls previous 3h	AvgDc
Average calling times previous 3h	AvgCount
Average cost previous 3h	AvgCost
Number of distinct calls	Dc
Destination countries	Countries
Calling number	ANumber
Number of calls	Count
Call duration	Duration
Call cost	Cost
Time	Time

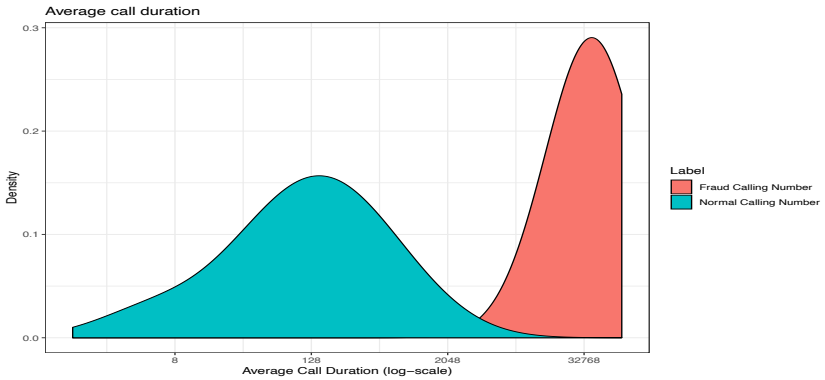

Fig. 2.2 Average Call Duration coloured by ground truth

The dimensionality of D is 64000×10 and in Table 2.1 these 10 attributes are listed. In addition, POST's experts have manually labeled all the data points of the dataset as fraudulent and non-fraudulent. The fraudulent calls have been confirmed as fraudulent whereas the non-fraudulent are not necessarily non-fraudulent, but may contain previously unnoticed types of fraudulent calls. All call activities were made across a time span of one month. There are 40930 unique calling numbers (ANumber) in D and only 0.04% of them have at least 1 fraudulent call.

Henceforth, we refer to a *ANumber* (calling number) whose calling activity is comprised of at least 1 fraudulent call as a *fraud calling number*. Conversely, we refer to a *ANumber* (calling number) whose calling activity is comprised of zero fraudulent call as a *normal calling number*. The overall percentage of fraudulent calls is 0.57% which makes D significantly imbalanced and suitable for developing unsupervised outlier detection techniques. Additionally, the authors in [208] provide a set of datasets that have been widely used in the outlier detection literature. The outlierness percentage of these datasets varies significantly between 0.03% and 32% whereas the number of data points varies between 129 and 567479 data points.

2.3.2 Exploratory Data Analysis

In this section, an Exploratory Data Analysis (EDA) is performed tto develop an understanding of our data. The easiest way to perform an EDA is to ask questions and then focus on the appropriate part of the dataset that will helps us to answer the questions. This type of analysis is crucial to find the most suitable machine learning model. As such, Fig. 2.3 illustrates the total number of individual call that are made exclusively by the *fraud call numbers*; 12 in total. Additionally, this figure helps us to visualise the class imbalance of our fraud detection

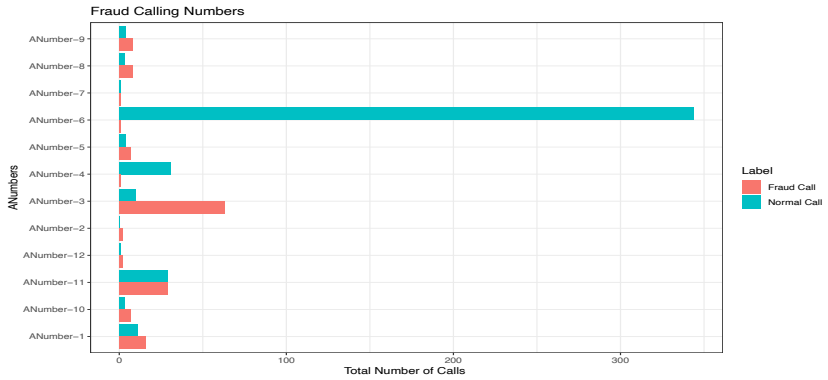

Fig. 2.3 Bar-plot of all Fraud Calling Numbers

problem. The fact that *fraud call numbers* produce a considerable amount of normal calls makes our problem highly imbalanced.

Moreover, in Fig. 2.2 the density of average call duration for all the ANumbers is visualized. The motivation of generating this plot is to try to answer the question: *Is there any difference between the Fraud Calling Numbers and the Normal Calling Numbers regarding the call duration distribution?*. Fig. 2.2 illustrates a clear difference between fraudsters and legitimate customers which helps us to conclude that fraud calls last longer than the normal calls.

2.3.3 Feature Engineering

In an unsupervised setting the ground truth is missing in order to find the best performing set of features. Hence, domain knowledge is significantly important to translate this knowledge into informative features that also will not be noisy. As a result, we take advantage of POST's domain knowledge aiming to expose more informative patterns. More specifically, two groups of features are calculated (i) Time agnostic features (i) Time dependent features.

Time agnostic features

- Month, Day, Weekday, Hour
- Continent percentage: What fraction of *countries* called belong to Europe, Africa, Asia, etc.?
- What is the difference of continent percentage between 2 calling events from the same *ANumber*?
- Duration difference of two consecutive calls (Calculated by *ANumber* and *ANumber* & *Dc*)

- How much time passed between two calls? (Calculated by *ANumber*)
- How many times called distinct calling numbers
- Duration per distinct call
- Cost for each Continent
- Average cost per call
- Average cost per destination calling number
- Count of call for each Continent
- Call duration duration by Continent

The above composite features have been calculated by ignoring the time variable. Hence, we calculate features that are used when time series problems are transformed into machine learning problems. More specifically, lag features are able to reveal time dependent patterns. Below, the calculated lag features are described.

Time dependent features

- Rolling Average of *Duration* for the last 3 calls made by each *ANumber*
- Rolling Average of *Duration* for the last 3 calls made by each *ANumber* per hour of the day
- Rolling Average of *Cost* for the last 3 calls made by each *ANumber*
- Rolling Average of *Cost* for the last 3 calls made by each *ANumber* per hour of the day
- Rolling Average of *Dc* for the last 3 calls made by each *ANumber*
- Rolling Average of *Dc* for the last 3 calls made by each *ANumber* per hour of the day

In the rest of this chapter, S is the high-dimensional dataset which is comprised of the attributes of D and all the above handcrafted features. More formally,
$S = D \cup \textit{Time dependent features} \cup \textit{Time agnostic features}$. The dimensionality of S is equal to 64000×91, where all the 91 attributes are numerical. In addition, it should be noted that the construction of the ensembles was developed using parallel computer architecture to improve efficiency.

2.3.4 Hardware & Software

The experiments were performed in the R and Python programming languages. Also, a 2.4 GHz Intel Xeon E5, 50 GB RAM, running Ubuntu 16.04 machine was used to carry out all the experiments.

Furthermore, the R and Python programming languages were used to conduct the experiments. R was used for the feature engineering part with the *data.table* package [73] and implemented KDEOS and LoOP detectors by using the *DDoutlier* package [174]. The iForest detector implemented using Python's scikit-learn library [199].

2.3.5 Evaluation Measures

ROC curve is the most widely-used evaluation measure in unsupervised outlier detection. This curve is obtained by plotting all possible true positive rates versus all possible false positive rates. Furthermore, the area under the ROC curve (AUC) measure summarizes the ROC curve by a single value. The perfect AUC score is equal to 1, whereas an inverted perfect AUC score is equal to 0. In our analysis, the AUC evaluation measure is used to evaluate the unsupervised outlier detectors.

Frequently, the user of a method is only interested in a small subset of the results that consists of the top-ranked objects (most anomalous data objects). Hence, the number n should be specified in advance which represents the n top-ranked objects. Afterwards, the precision at n (P@n) van be calculated. More formally, for a dataset D of size N, consisting of outliers O $\subset$ D and inliers I $\subseteq$ D (D = O $\cup$ I), P@n can be formalized as:

$$P@n = \frac{|o \in O | rank(o) \leq n|}{n} \tag{2.1}$$

In our analysis, the first n = 400 events with the greatest outlier score are used as the top-ranked objects in order to calculate the precision measure; P@400. The number 400 is selected based on our intuition regarding the outlierness percentage of the dataset.

2.4 Results and Discussion

In Fig. 2.4, 2.5, 2.6, 2.7 box plots are presented that are related to AUC performance of all the bagging executions of all detectors. The only difference between these figures is the normalization scheme that is followed. More specifically, the variance in the box plots is produced due to the randomness between all the executions of the detectors during the bagging procedure. Each box plot summarizes 100 different executions of a base detector

(KDEOS, LoOP, iForest) created by different parameter values and random projections to lower dimensions.

Fig. 2.8 produced by combining the results of each detector in Fig. 2.4, 2.5, 2.6, 2.7 with the average or the maximum combination function. For instance, the top left plot in Fig. 2.8 is produced when the outlier scores produced by LoOP,KDEOS and iForest using the feature bagging technique are combined with the average and maximum combination function. The outcome of this procedure is an unsupervised outlier ensemble for each detector and each combination function. Overall, Fig. 2.8 presents all possible *Bagging Ensembles* of our experimental analysis as discussed in Sec. 2.2.7.

LoOP detector performs slightly better than random guess and iForest shows the highest values of the AUC. iForest managed to detect all the *fraud calling numbers*.

In addition, Table 2.2 and 2.3 show the standard deviation of AUC for *Bagging Ensembles* with two different levels of aggregation. More specifically, Table 2.2 presents the standard deviation of the *Bagging Ensembles* constructed with either the average or maximum combination function across all the normalization schemes. In other words, Table 2.2 demonstrates how robust is an ensemble, constructed with the maximum or average function, to the different normalization schemes. Additionally, Table 2.3 presents the standard deviation of *Bagging Ensembles* regardless the normalization scheme and combination function. In other words, in 2.2 is presented how robust is each detector to all possible combinations to construct a *Bagging Ensemble*.

Overall, the *Bagging Ensemble* constructed with the KDEOS algorithm and either the average or maximum combination function demonstrates the largest standard deviation. In addition, the KDEOS outlier detection algorithm demonstrates the largest standard deviation to all possible combinations to construct a *Bagging Ensemble*. Hence, the KDEOS detector is the most influenced algorithm by data normalization schemes and combination functions. On the other hand, the iForest algorithm demonstrated the most robust performance across different normalization schemes and/or combination functions.

Table 2.2 Standard deviation of AUC values across all normalization schemes for each Bagging Ensemble

Detector	**Combination Function**	**Std.**
iForest	Maximum	**0.0001**
	Average	**0.0001**
KDEOS	Maximum	0.1718
	Average	0.1116
LoOP	Maximum	0.0494
	Average	0.0626

Table 2.3 Standard deviation of AUC values across all the normalization schemes and all combination functions for all the Bagging Ensembles

Detector	**Std.**
iForest	**0.0001**
KDEOS	0.1555
LoOP	0.0755

One key benefit of outlier ensemble learning is the ability to take advantage of diverse ensemble members (outlier detection algorithms) in order to construct better performing detectors than the individual ensemble members. Especially well-established subspace techniques induce diversity by inducing randomness in the resulting models and make the ensemble perform better. An example of the improvement achieved by a good outlier ensemble is the KDEOS detector of our analysis. In particular, Fig. 2.4 illustrates that this detector do not manage to obtain a single execution with AUC higher than 0.7. However, Fig. 2.8 presents that KDEOS *Bagging Ensembler* constructed with the average combination function produces AUC values higher than higher than 0.8.

Furthermore, the *Bagging Ensemble* that uses the LoOP detector improved itself the least compared to its individual executions. The LoOP detector achieves the best performance when both the Median-MAD normalization scheme and the average combination function are applied. In addition, the iForest algorithm is the only detector that is not affected at all by data normalization schemes and combination functions. It steadily shows performance close to the perfect; AUC values equal to 1. One possible explanation is that this detector is an ensemble by its design compared to KDEOS and LoOP and that probably leads more accurate and robust predictions.

In addition to the AUC evaluation measure, the P@N (Precision at n) evaluation measure is used. Fig. 2.9 presents the Precision (P@n) results of (i) all the individual *Bagging Ensembles* (ii) the combination of all the *Bagging Ensembles* using α) the average combination function, β) the maximum combination function. Hence, Fig. 2.9 is vertically splitted into two parts where on the left the maximum function is used and on the right the average. Furthermore, Fig. 2.9 illustrates that that the combination of all the *Bagging Ensembles* using either the average or maximum function improves the performance of each individual *Bagging Ensemble* except of iForest; the best performing algorithm. The average combination function outperforms the maximum combination function at the three out of four normalization schemes; except for the Min-Max normalization.

2.5 Conclusion

Constructing outlier ensembles on high-dimensional data is challenging and this paper highlights the difficulty in selecting the best core components of an outlier ensemble pipeline. Addressing a real-world problem with unsupervised techniques requires overcoming these challenges to obtain both robust and accurate predictions. Researchers often develop novel unsupervised methods in artificial environments using toy data sets and therefore do not need to analyze the sensitivity of their approach. In contrast, problems encountered by companies need to address the problem of results varying significantly in order to deploy a robust and reliable solution based on these methods.

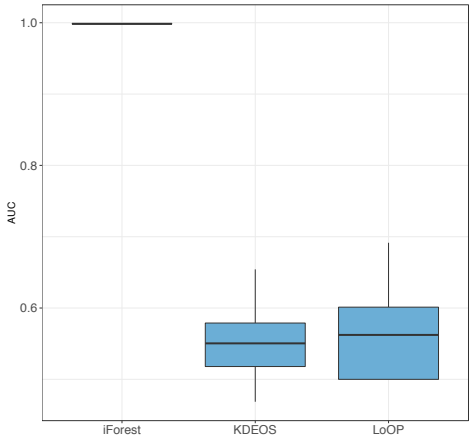

Fig. 2.4 Feature bagging variability of AUC of KDEOS, LoOP and iForest on Mean-SD normalization

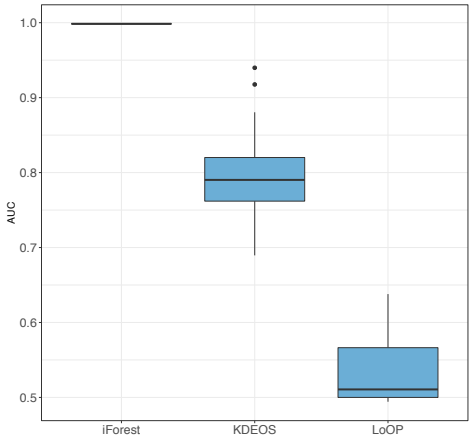

Fig. 2.5 Feature bagging variability of AUC of KDEOS, LoOP and iForest on Min-Max data normalization

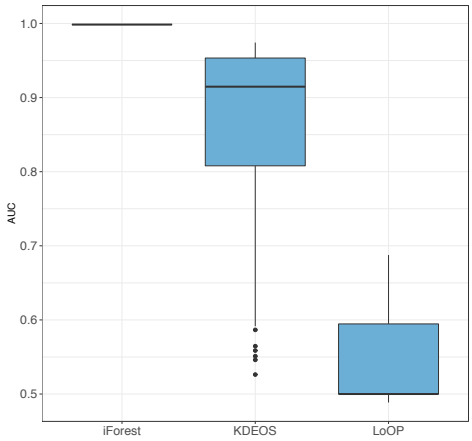

Fig. 2.6 Feature bagging variability of AUC of KDEOS, LoOP and iForest on Median-MAD normalization

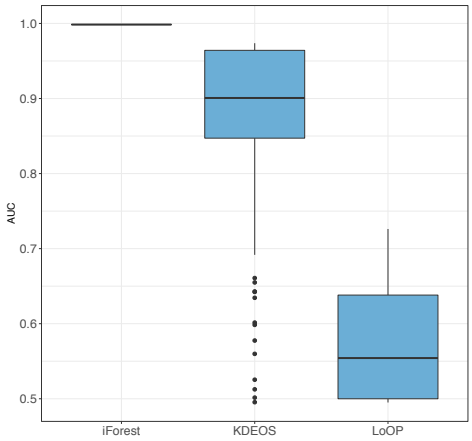

Fig. 2.7 Feature bagging variability of AUC of KDEOS, LoOP and iForest on Median-IQR normalization

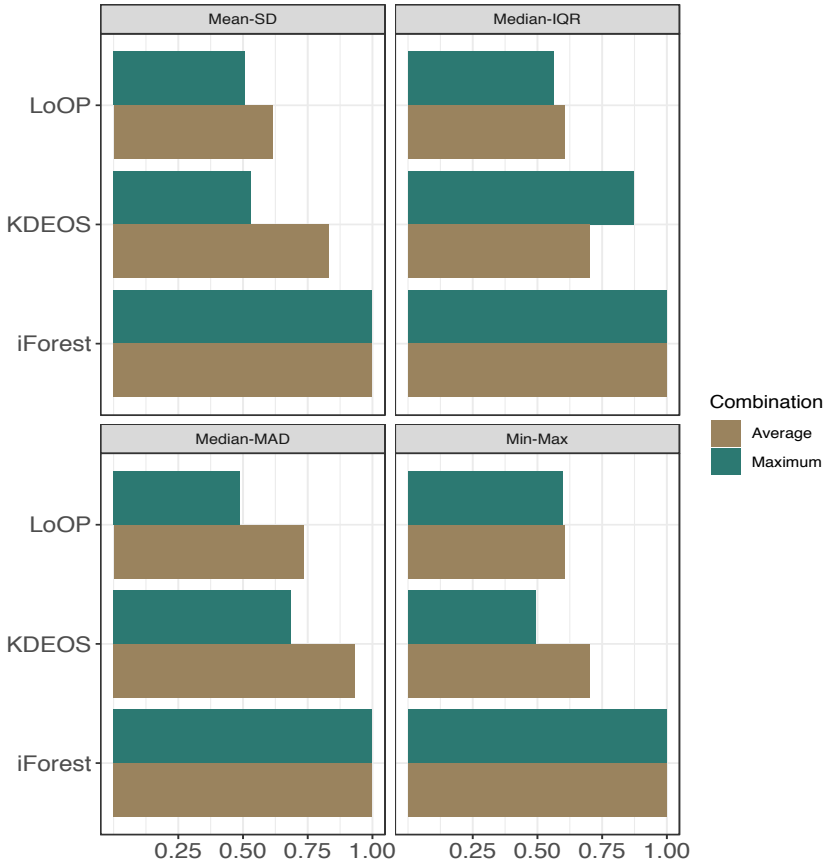

Fig. 2.8 AUC Performance of all the *Bagging Ensembles*

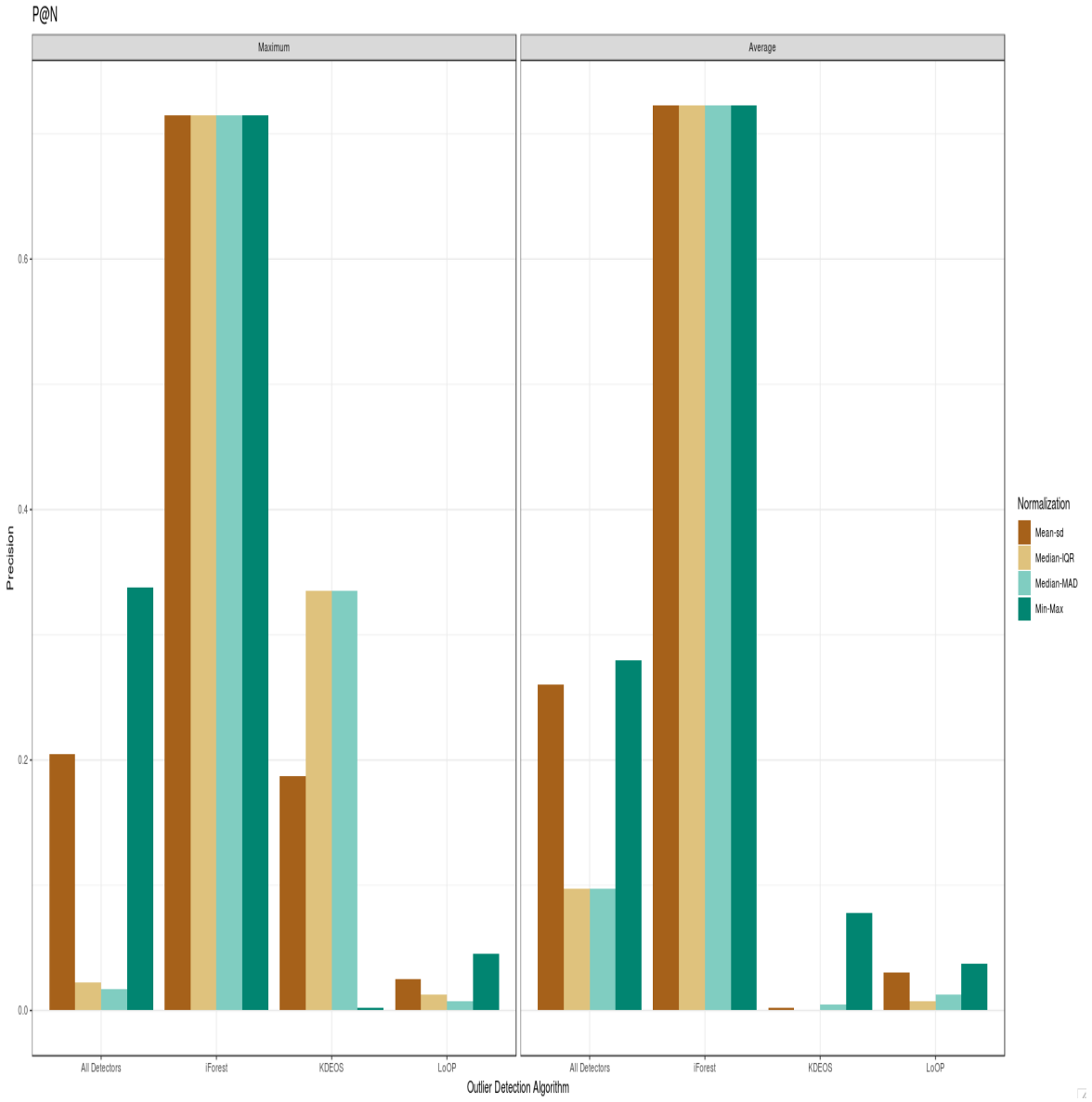

Fig. 2.9 Precision (P@400) of all the *Bagging Ensembles* and their combination. On the left, the Maximum combination function is used for iForest, KDEOS, LoOP, and, the ultimate combination of all Bagging Ensemblers. On the right the Average combination function is used.

Part III

Application Domain Agnostic Novelty Detection

Chapter 1

OCC Ensembles with Unsupervised Representations to Detect Novelty

Supervised anomaly detection approaches learn from already classified objects whereas unsupervised anomaly detection algorithms score data according to their exceptionality. Recently, improvement has been demonstrated in the problem of supervised anomaly detection by leveraging the strengths of both worlds; supervised learning and unsupervised learning. Supervised learning approaches demand sufficient labelled training sets but usually the knowledge related to the anomalous class is not sufficient. More specifically, in many anomaly detection scenarios training data are available and describe objects belonging to a particular class (usually normal objects), but very little data (if any) describing objects that do not belong to this class (abnormal objects).

In this chapter, we address the problem of novelty detection with the aid of one-class classification learners to accommodate the scarcity of sufficient labelled training sets. In particular, a framework is introduced that first leverages the strengths of unsupervised scoring algorithms to learn new data representations and afterwards develops two one-class classification ensembles to detect novelty. The introduced method is the first attempt to detect novelty with one-class classification ensembles developed on unsupervised representations.

1.1 Introduction

A novelty can be considered as a specific type of anomaly that does not fit well with the previously learned distributions [202]. Novelty detection is of paramount importance to real world applications such as credit card abuse detection in financial transactions data or the identification of fraudulent calls in telecommunication data. Additionally, the problem of

novelty detection is also known as one-class classification [242] where data from one class, the target class, is used during the learning phase. The task is to define a boundary around this class to minimize the chance of accepting anomaly or outlier objects. Afterwards, the learnt model is used to classify if an unseen observation belongs to the target class or not. It is worth noting that, it is hard to decide on the basis of just one class how tightly the boundary should fit in each of the directions around the data. Throughout this chapter the terms *"anomalies"* and *"outliers"* [114] are used interchangeably.

In the absence of labelled training data, unsupervised outlier detection algorithms or unsupervised detectors are the most suitable techniques [283, 10]. These detectors assign to each object a score reflecting its "outlierness" and the study in [45] extensively evaluates a plethora of unsupervised detectors which are based on neighborhoods in Euclidean space. Recently, Micenková et al. [183] for the first time leveraged the strengths of unsupervised outlier scoring detectors to improve supervised outlier detection. Specifically, unsupervised scoring detectors learn new feature representations that can be used to augment the original data and improve anomaly detection performance in a fully supervised setting. In the same spirit of the best of both worlds (supervised and unsupervised), the authors in [274, 184, 49] have introduced different variants of this approach, all of which address the supervised outlier detection problem.

This work investigates the improvement of novelty detection by introducing a framework composed of **O**ne-**C**lass **C**lassification (OCC) ensembles developed on enriched unsupervised representations. Additionally, this work is motivated by: 1. the authors of [242] who suggest one-class classifiers as the preferable approach to address scenarios where few outliers are known, 2. Micenková et al. [183] who first leveraged the strengths of unsupervised outlier scoring detectors to improve the anomaly detection rate. As such, firstly well-studied outlier detection algorithms [45] are used from the Knowledge Discovery in Databases (KDD) field to learn unsupervised representations. Afterwards, these unsupervised representations are leveraged in two different ways to build two one-class classification ensembles. Finally, these ensembles are randomly instantiated and compared against the OCC learner developed on the original feature representation. In addition, we make the following contributions:

- We address the one-class classification problem compared to existing works that leverage unsupervised representations to detect anomalies.
- We propose a novel strategy regarding how to build one-class classification ensembles to accommodate the inability of OCC learners to perform feature selection in contrast to supervised classifiers.

- We perform a more comprehensive experimental investigation by analyzing 175 datasets from the repository [45] to thoroughly evaluate our framework. We increase 14.5 times the number of datasets of the most extended supervised-learning work [184].
- In addition, our framework's ensembles demonstrate statistically significant improvement compared to existing works that leverage unsupervised representations to detect anomalies; they fail to demonstrate a statistically significant performance improvement.

To the best of our knowledge, this paper introduces for the first time a framework to detect novelties in an one-class classification setting by learning effectively unsupervised representations to eventually build two one-class classification ensembles.

The rest of the paper is organised as follows. We introduce our method in Sect. 1.2 and we evaluate our method on a large collection of outlier datasets in Sect. 1.3. Finally, we present the experimental results and conclude the paper in Sect. 1.4 and Sect. 1.5 respectively.

1.2 One-Class Classification Ensembles with Unsupervised Representations

In this section the design of our framework is described. First, it **learns unsupervised representations** by using well-studied outlier detection methods from the Knowledge Discovery in Databases field, such as LOF [39] and LoOP [145]. Afterwards, it **constructs two one-class classification ensembles** by inducing diversity in ensemble models [279]. Finally, it combines classification predictions on random representations to eventually produce a global classification target vector.

1.2.1 Unsupervised Representation Learning

We select unsupervised outlier detection algorithms Ω_i that have the ability to output a scoring vector. This vector is influenced by the detector's bias and describes the degree of outlierness. More formally, when a detector Ω_i is employed on a dataset D with n data objects, it produces a real valued scoring vector $\Omega_i(D) \in \mathbb{R}^{n\times 1}$. Throughout this work, this scoring vector is considered as an one-dimensional unsupervised representation of a given dataset D.

Let $X \in \mathbb{R}^{n\times k}$ denote the original feature space of a given dataset D. We employ a set of heterogeneous unsupervised outlier detection algorithms, $\Omega = \{\Omega_1, \Omega_2, ..., \Omega_m\}$, where m is the total number of detectors. An unsupervised representation of X is learnt by applying Ω on X. Different biases and notions of outlierness from different detectors are captured by

using a collection of heterogeneous unsupervised detectors Ω. We refer to the unsupervised representation of X as **U**nsupervised **F**eature **S**pace (*UFS*):

$$UFS = [\Omega_1(X), \Omega_2(X), ..., \Omega_m(X)] \in \mathbb{R}^{n \times m} \tag{1.1}$$

Working with heterogeneous detectors emerges a problem related to the fact that different algorithms differ widely in their scale and meaning. The authors in [146] provide a unification approach that translates arbitrary outlier scoring vectors to values in the range $[0,1]$. We follow normalization techniques introduced in their study to re-scale all set members of Ω and in subsection 1.3.2 further details are provided.

In addition to the unsupervised representation of X, we construct an augmented version of the original feature space X by leveraging the already produced *UFS*. In particular, we refer to the augmented representation of X as **A**ugmented **F**eature **S**pace (*AFS*) which is defined as it follows:

$$AFS = [X, UFS] \in \mathbb{R}^{n \times (k+m)} \tag{1.2}$$

The augmented version of the original feature space X was firstly proposed by the authors in [183].

1.2.2 Construction of the One-Class Classification Ensembles

The core idea of how we build our ensembles is aligned with methods for constructing outlier ensembles [279, 13], classification ensembles [40], or clustering ensembles [91]. The authors in [279, 13], extensively discussed the importance of having diverse ensemble members to build good ensembles. In this study, we induce diversity by introducing randomness. In particular, from equations (1.1) and (1.2) we observe that a different parameterization of at least one Ω_i is sufficient to lead to a different realization (instance) of *UFS* and, consequently, of *AFS*. We formally define below what such random realizations are.

Definition 6 *UFS-RR: An UFS-Random Realization is a random unsupervised representation produced by randomly assigning user-defined parameter values to all $\Omega_i \in \Omega$.*

Definition 7 *AFS-RR: An AFS-Random Realization is a random augmented representation produced when we concatenate the original feature vectors of X with a UFS-RR.*

As a next step, we exemplify what an *UFS-RR* and *AFS-RR* is:
Let $\Omega = [KNN(\Phi = 2), LOF(\Phi = 99), LoOP(\Phi = 48)]$ be a collection of unsupervised detectors composed of the algorithms KNN [206], LOF [39], and LoOP [145] employed

Algorithm 2

Input: *URL* := a set of *w UFS-RR*, *ARL* := a set of *w AFS-RR*; $w \in \mathbb{N}$

Output: OS_{URL} := Global Classifications of *URL*, OS_{ARL} := Global Classifications of *ARL* (both binary column vectors)

1: **for** i = 1 to *w* **do** ▷ w is the cardinality of *URL* and *ARL*
2: *data* := URL_i or *data* := ARL_i
3: *train data* := 80%
4: *hold-out data* := 20%
5: employ OCC on *train data* ▷ Learn optimal parameter values ith K-fold Cross Validation for an One-Class Classifier (OCC)
6: OS_i := outlier scores of *hold-out data* ▷ Predict with optimal parameter values
7: OS := [OS_1, OS_2,..,OS_w] ▷ Matrix of *w* outlier score vectors
8: Outlier Classification of each $OS_i, i \in \{1,2,..,w\}$ ▷ OS_i is a binary vector
9: $OS_{URL} := GlobalClassification(OS)$ or $OS_{ARL} := GlobalClassification(OS)$

with parameter values $\Phi = 2$, $\Phi = 99$, and $\Phi = 48$, respectively. All these algorithms take as input only one parameter (Φ represents the neighbourhood size) which is randomly drawn from a uniform distribution $\mathscr{U}[1,100]$. Next, by employing Ω on $X \in \mathbb{R}^{n \times k}$, an *UFS-RR* $\in \mathbb{R}^{n \times 3}$ and *AFS-RR* $\in \mathbb{R}^{n \times (k+3)}$ are produced. Overall, our novelty detection framework is composed of two ensembles developed on the following representation spaces:

1. $Ensemble_{UFS}$ is developed on multiple *UFS-RR*
2. $Ensemble_{AFS}$ is developed on multiple *AFS-RR*

In particular, we construct our ensembles by following the steps of Algorithm 2. Let *w* be a user defined parameter which reflects the number of different *UFS-RR*. As such, it receives as input, *w UFS-RR* to create the set *URL* (**U**nsupervised **R**epresentation **L**earning) and *w AFS-RR* to create the set *ARL* (**A**ugmented **R**epresentation **L**earning) (i.e. $|URL| = |ARL| = w$). Afterwards, for each set member of *URL* and *ARL*, which are considered as datasets in the following steps, we employ a K-fold cross-validation procedure as in [242, 241, 236, 119, 121] to estimate the performance of the OCC algorithm (Section 1.3 provides details regarding the exact procedure). Finally, by classifying new data in the hold-out set, *w* OCC outlier score vectors are produced by the *URL* set and *w* OCC outlier score vectors are produced by the *ARL* set.

In our framework the *w* parameter has more weight regarding the predictive capabilities compared to the cardinality of the Ω set. In other words, we rely more on the number of the ensemble members that our one-class classification ensembles are composed of, instead of the total number of the heterogeneous detectors in Ω. As such, we overcome the

inability of OCC learners to perform feature selection by keeping low the cardinality of the Ω set in order not to considerably affect the dimensionality of the original feature space X.

Outlier Classification: For the sake of simplicity we explain the remaining steps for the w OCC outlier score vectors produced by the *URL* set. It is worth noting that each of w OCC outlier score vectors are real valued; $w_i \in \mathbb{R}^{n\times 1}$. Combining individual classifications to a final prediction is a vital step to construct an ensemble and we are inspired by the authors in [248] who combine classifications from different subspaces to produce a final prediction. Hence, we transform all the w OCC outlier score vectors to binary vectors based on a threshold influenced by the OCC's design. For instance, outlier scores produced by the One-Class SVM [223] algorithm represent the signed distance to the separating hyperplane where the distance is positive for an inlier and negative for an outlier. In addition, outlier scores produced by the SVDD algorithm [243], represent the distance to the decision boundary where it is positive for an outlier observation. Henceforth, we call as *Outlier Classification* the outcome of this transformation procedure where outliers and inliers take the values of 1 and 0 respectively. Finally, we concatenate all the w *Outlier Classifications* (binary vectors) to create a matrix called OS $\in \mathbb{R}^{n\times w}$.

Final Target Vector: The vectors $OS_{URL} \in \mathbb{R}^{n\times 1}$ and $OS_{ARL} \in \mathbb{R}^{n\times 1}$, are created by combining w *Outlier Classifications* (binary vectors). The combination is achieved by defining what a global outlier and inlier is. As such, by following Definition 8 an observation x is a global inlier if it is an inlier in all *Outlier Classifications*. Additionally, an observation x is a global outlier if x is an outlier in at least one *Outlier Classification*. Definition 8 is inspired by the authors in [248] who combine classifications from different subspaces to produce a final prediction.

Definition 8 *(Global Classification) Let a set of w Outlier Classifications $OS_1, \ldots, OS_w$ be given. A global classification for these Outlier Classifications is a function*

$$f(x) = \begin{cases} 1, if \sum_w OS_w(x) > 0 \\ 0, otherwise \end{cases}$$

Finally, Algorithm 2 lists all the above steps in pseudo code and Fig. 1.1 visually illustrates our framework. It is worth noting that steps 1 and/or 5 of Algorithm 2 could be implemented by following parallel computing approaches for speedup.

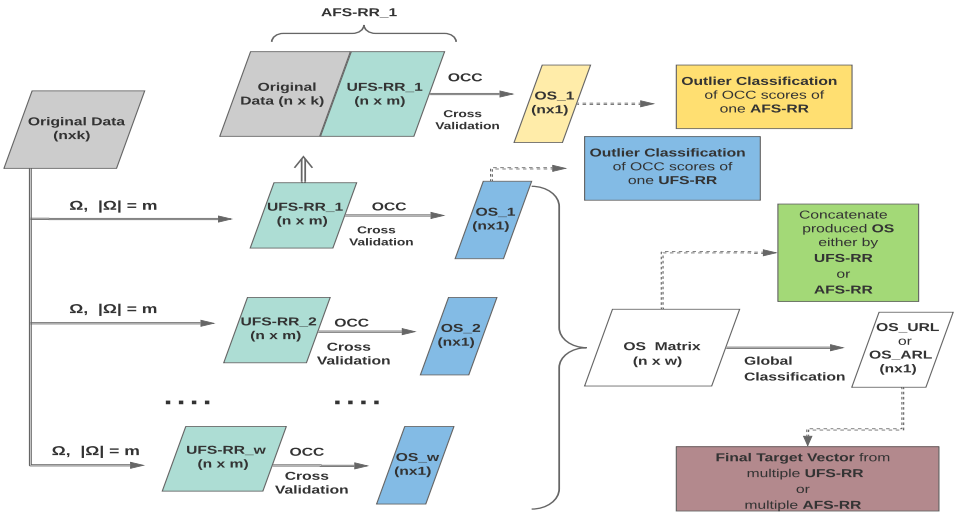

Fig. 1.1 One-Class Classification (OCC) Ensembles of our framework

1.3 Experiments and Evaluation

In this section, a set of experiments is developed to investigate whether our framework's one-class classification ensembles outperform baseline approaches. This investigation will help us to answer the research question: *"Does incorporating unsupervised representations by following our proposed learning algorithm improve novelty detection?"*.

1.3.1 Datasets

A benchmark data repository for outlier detection [45] that is composed of 23 basic datasets was used to perform our experiments. These datasets are processed in different ways to provide variants with different percentage of outliers, different handling of dataset characteristics such as duplicates, attribute normalization, and categorical values. As suggested in [45], we select the normalized without duplicates version of datasets. In addition, where applicable, we select the 10 versions with the smallest rate of outliers. Overall, for our analysis we use 175 datasets which come from 22 basic datasets and Table 1.1 presents their characteristics. The KDDCup99 dataset was excluded from our analysis due to its large size that more than severely affected training time [30]. Figure 1.2 illustrates the outlier percentage and the number of attributes of the datasets coloured the category they represent.

The datasets in the repository [45] can be divided into two broad categories: 1. semantically meaningful datasets where the vast majority is related to medical applications, 2. datasets that have appeared in the outlier detection literature. Table 1.1 is a summary of the datasets in our experiments. Furthermore, almost 93% of the datasets in our analysis (163

Table 1.1 Datasets characteristics

Name	Versions	Observations	Attributes	Outliers	Percentage	Category
ALOI	1	50000	27	1508	3.04%	Literature
Glass	1	214	7	9	4.21%	Literature
Ionosphere	1	351	32	126	35.90%	Literature
Lymphography	1	148	3	6	4.05%	Literature
PenDigits	10	9868	16	20	0.20%	Literature
Shuttle	10	1013	9	13	1.28%	Literature
Waveform	10	3443	21	100	2.90%	Literature
WBC	10	223	9	10	4.48%	Literature
WDBC	10	367	30	10	2.72%	Literature
WPBC	1	198	33	47	23.74%	Literature
Annthyroid	10	6729	21	134	1.99%	Semantic
Arrhythmia	10	248	259	4	1.61%	Semantic
Cardiotocography	10	1681	21	33	1.96%	Semantic
HeartDisease	10	153	13	3	1.96%	Semantic
Hepatitis	10	80	19	13	4.29%	Semantic
InternetAds	10	1630	1555	32	1.96%	Semantic
PageBlocks	10	4982	10	99	1.99%	Semantic
Parkinson	10	53	22	5	9.43%	Semantic
Pima	10	510	8	10	1.96%	Semantic
SpamBase	10	2579	57	51	1.98%	Semantic
Stamps	10	315	6	16	1.90%	Semantic
Wilt	10	4655	5	93	2.00%	Semantic

out of 175) contain less than 5%. In addition, the datasets are diverse in terms of number of attributes and the application scenarios. Hence, based on the aforementioned facts we ensure that our proposed framework is more than well tested.

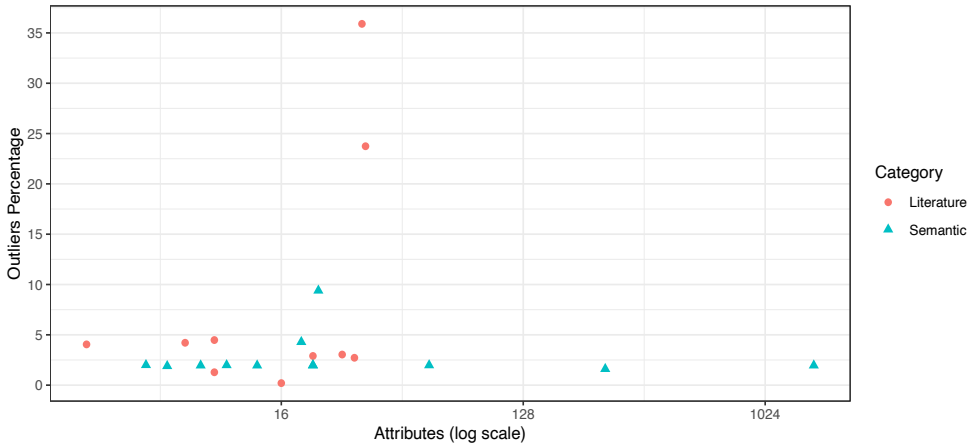

Fig. 1.2 Scatter plot of the datasets characteristics

1.3.2 Experimental Setup

Reproducibility

Our experiments were implemented in R and Python using the freely available libraries[1]. Also, all experiments were carried out on a 2.4 GHz Intel Xeon E5, 50 GB RAM, running

[1]R packages: reticulate, data.table, Python libraries: scikit-learn

Ubuntu 18.04. The corresponding Github code is freely available on URL[2].

Unsupervised Representation Learning:

12 neighborhood-based unsupervised outlier detection algorithms studied in [45], are used to learn the unsupervised representations. In particular, for each detector a parameter value, which represents the neighborhood range, is randomly drawn from a uniform distribution $\mathscr{U}[1,100]$.The bounds of this uniform distribution are studied in [45]. Instead of re-calculating the outlier scores, we leverage the freely available outlier score results[3] provided by [45]. The outlier detection algorithms of our experimental setting are: KNN [206], KNNW [21], LOF [39], SimplifiedLOF [227], LoOP [145], LDOF [272], ODIN [104], FastABOD [148], KDEOS [226], LDF [157], INFLO [124], and COF [237]. The description of these algorithms is omitted due to brevity. Henceforth, Ω is composed of all the aforementioned outlier detection algorithms, $|\Omega| = 12$.

Furthermore, in order to create the *UFS-RR* and *AFS-RR* it is important to normalize all $\Omega_i \in \Omega$. Hence, the min-max normalization which is a linear transformation is applied to transform the outlier scores of all $\Omega_i \in \Omega$ into the interval $[0,1]$. The authors in [146] suggest that for the ABOD (FastABOD $\in \Omega$) algorithm it is important to do an inversion and to stretch the outlier scores which are concentrated at very small values(for details see section 3.2.3 of [146]). As such, the logarithmic inversion is applied to ensure that before and after the inversion outliers keep their ranking. The logarithmic function is monotone, and as a consequence the inversion is ranking-stable. Finally, on top of the logarithmic inversion, the min-max normalization is applied to transform the outlier scores into the interval $[0,1]$.

One-class Classifier

We employ one of the most popular one-class classifiers, the One-class SVM (OCSVM) [223]. The motivation of this work is not to compare different OCC algorithms thus we employ a state-of-the-art one one-class learner to focus on evaluating the novelty detection improvement. It is worth noting that our framework's ensembles are able to be constructed with any OCC algorithm.

The authors of the OCSVM algorithm [223] prove that the Gaussian kernel has the advantage of always separating the data from the origin. In addition, they conclude that $\nu \in (0,1]$ and γ are the suitable parameters to tune. Hence, in our experiments we use the Gaussian kernel and we optimize the ν and γ parameters.

[2]URL will be included on paper release due to double blind review process

[3]The authors used the ELKI [225] public software

Learning Scenario

One-class learners do not have any mechanism to use label information directly to train the decision function. Instead, they could learn this decision function by being exposed to training data composed of labeled inliers only or leverage negative examples $SVDD_{neg}$ [243], SSAD [96]. The final goal of a one-class classifier is to find a generalizable solution by minimizing the false positives and false negatives. Unfortunately, false negatives cannot be estimated when no outlier objects are available. To accommodate this issue the authors in [242] proposed different ways to generate artificial[4] outliers. In this study, we employ one-class classifiers exposed to inliers during the training phase. Also, we minimize the false negatives by using limited knowledge from the true outlier distribution. We purposely avoid investigating different ways of generating artificial outliers because it would add significant complexity to our attempt to answer the research questions of this work. It should be emphasized that our proposed methodology remains identical in cases that outliers need to be artificially generated.

Evaluation

We follow a K-fold cross-validation [Kohavi et al.] procedure as in [242, 241, 236, 119, Janssens and Postma] to estimate the performance of the OCSVM algorithm applied on each *UFS-RR* or *AFS-RR* data set. We split the *UFS-RR* and *AFS-RR* data sets using the popular 80%-20% rule. As such, a hold-out set containing 20% randomly selected data points of the entire data set is reserved. With the remaining 80% a 10-fold cross-validation procedure is applied to optimize the parameters of the OCSVM. Afterwards, the OCSVM is trained with the optimal parameter values obtained by the cross-validation with respect to ROC-AUC. We recall that the OCSVM is trained only on the normal data points; the training set consists of nine folds and the test consists of one fold. To minimise the bias introduced by the random selection of folds, we repeat the 10-fold cross-validation 30 times. Finally, an OCC outlier score vector for a *UFS-RR* or *AFS-RR* is produced and its size is equal with the size of the hold-out subset. This vector is considered as an ensemble member of the $Ensemble_{UFS}$ or $Ensemble_{AFS}$.

The output of the $Ensemble_{UFS}$ or $Ensemble_{AFS}$ is a binary column vector regardless of the number of ensemble members are composed of (see Algorithm 2 and section 1.2.2 for details). Hence, we evaluate the novelty detection performance by evaluating this binary column vector. Recall at N (R@N) and Precision at N (P@N) [13, 45], are widely used for evaluation in outlier detection and are appropriate for binary predictions. A rank list of the outliers is needed for both measures in order to evaluate the top-N outliers that are the first N

[4]The corresponding MATLAB toolbox [241]

in the ranking. $N = |O|$ where O is the number of outliers presented in the hold-out set is a widely used choice in order to evaluate the detection performance on all the outliers. In this work, R@N and P@N are evaluated at $N = |O|$. For each basic dataset we average across the 10 versions (where applicable). Finally, we repeat 30 times this procedure and report the average P@N and R@N values of the 30 repetitions.

To analyze the performance differences among multiple algorithms we use the non-parametric Friedman test followed by Nemenyi post-hoc test[5] as proposed in [69]. We consider $p < 0.05$ to be statistically significant. In addition, in [69] the author introduces the *critical difference plot* to check visually the differences. For a significance level α the test determines the critical difference (CD) in order to assess the difference between the average ranking of two algorithms.

Competitors

We generate multiple random instances of our framework's ensembles, $Ensemble_{UFS}$ and $Ensemble_{AFS}$, which are composed of w = [5, 10, 15, 20, 25, 30] *UFS-RR* and *AFS-RR* respectively. As such, following are the competitors: (i) *UFS-5RR* the $Ensemble_{UFS}$ composed of 5 *UFS-RR* , (ii) *UFS-10RR*, (iii) *UFS-15RR*, (iv) *UFS-20RR*,(v) *UFS-25RR*, (vi) *UFS-30RR*, (vii) *AFS-5RR* the $Ensemble_{AFS}$ composed of 5 *AFS-RR*, (viii) *AFS-10RR*, (ix) *AFS-15RR*, (x) *AFS-20RR*, (xi) *AFS-25RR*, (xii) *AFS-30RR*, (xiii) *Original baseline model* is the OCSVM employed on Original data X.

1.4 Results and Discussions

[5] Both tests are freely available by the R package [42]

Table 1.2 R@N - Average of 30 trials and 10 versions (where applicable)

dataset	Original	AFS 30RR*	AFS 25RR*	AFS 20RR*	AFS 15RR*	AFS 10RR*	AFS 5RR*	UFS 30RR#	UFS 25RR#	UFS 20RR#	UFS 15RR#	UFS 10RR#	UFS 5RR#
ALOI	0.1358	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**	**0.1360**
Glass	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**	**0.0995**
Ionosphere	0.2640	**0.3173**	**0.3173**	**0.3173**	**0.3173**	**0.3173**	**0.3173**	0.3107	0.3093	0.3093	0.3053	0.3080	0.3040
Lymphography	0.0130	0.0795	0.0784	0.0764	0.0755	0.0681	0.0617	0.0930	0.0930	0.0927	0.1093	**0.1211**	0.0650
PenDigits	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Shuttle	0.1032	**0.1766**	**0.1766**	0.1667	0.1667	0.1522	0.1377	0.1728	0.1695	0.1676	0.1634	0.1548	0.1479
Waveform	0.0030	0.0995	0.0984	0.0964	0.0955	0.0881	0.0817	**0.1130**	**0.1130**	0.1127	0.1093	0.1001	0.0840
WBC	0.4336	0.5443	0.5489	0.5421	0.5115	0.5275	0.5544	0.5195	0.5118	0.5127	0.5128	**0.5912**	0.5778
WDBC	0.5206	0.7465	0.7457	0.7416	0.7465	0.7395	0.7284	**0.7745**	0.7734	0.7724	0.7705	0.7710	0.7678
WPBC	0.0296	0.048	0.048	0.048	0.048	0.048	0.048	**0.0555**	**0.0555**	**0.0555**	**0.0555**	**0.0555**	0.05185
Annthyroid	0.0188	0.0512	0.0493	0.0472	0.0423	0.0376	0.0279	**0.0760**	0.0755	0.0743	0.7003	0.0629	0.0512
Arrhythmia	0.4022	0.4422	0.4402	0.4389	0.4380	0.4343	0.4317	**0.4544**	**0.4544**	**0.4544**	**0.4544**	0.4518	0.4499
Cardiotocography	0.4244	0.5118	0.5118	0.5099	0.5118	0.5085	0.4976	**0.5159**	**0.5159**	**0.5159**	**0.5159**	0.5134	0.5139
HeartDisease	0.1255	0.1587	0.1576	0.1576	0.1565	0.1542	0.1477	**0.1613**	**0.1613**	0.1580	0.1561	0.150	0.1383
Hepatitis	0.4385	**0.4989**	**0.4989**	0.4972	0.4947	0.4922	0.4847	0.4980	0.4980	0.4980	0.4980	0.4922	0.4964
InternetAds	0.2356	0.5287	0.5129	0.5281	0.5281	**0.5301**	0.5179	0.5282	0.5104	0.5199	0.5179	0.5342	0.5052
PageBlocks	0.7261	**0.7274**	0.7271	0.7272	0.7269	0.7272	0.7267	0.7239	0.7234	0.7221	0.7215	0.7192	0.6971
Parkinson	0.190	0.3233	0.3233	0.320	0.320	0.3166	0.3083	**0.330**	0.3283	**0.330**	0.3216	0.3233	0.3050
Pima	0.1455	0.1787	0.1776	0.1776	0.1765	0.1742	0.1677	**0.1813**	**0.1813**	0.1780	0.1761	0.170	0.1583
SpamBase	0.0730	0.2856	0.2843	0.2852	0.2846	**0.2979**	0.2751	0.2962	0.2962	0.2962	0.2958	0.2958	0.2896
Stamps	0.2597	**0.3120**	0.3094	0.3094	0.3097	0.3041	0.2865	0.3077	0.3067	0.3061	0.3051	0.2991	0.2952
Wilt	0.0367	0.0416	0.0413	0.0385	0.0358	0.0287	0.0210	**0.0455**	0.0453	0.0453	0.0442	0.0410	0.0381

Table 1.3 P@N - Average of 30 trials and 10 versions (where applicable)

dataset	Original	AFS 30RR*	AFS 25RR*	AFS 20RR*	AFS 15RR*	AFS 10RR*	AFS 5RR*	UFS 30RR#	UFS 25RR#	UFS 20RR#	UFS 15RR#	UFS 10RR#	UFS 5RR#
ALOI	0.0415	**0.0416**	**0.0416**	**0.0416**	**0.0416**	**0.0416**	**0.0416**	0.0415	0.0415	**0.0416**	**0.0416**	**0.0416**	**0.0416**
Glass	**0.1462**	0.0801	0.0661	0.0716	0.0716	0.0626	0.0684	0.0667	0.0683	0.0667	0.0702	0.072	0.0616
Ionosphere	**0.9648**	0.8405	0.8405	0.8405	0.8405	0.8405	0.8405	0.8311	0.8311	0.8311	0.8311	0.8311	0.8311
Lymphography	0.0079	0.0108	0.0106	**0.0109**	0.0108	0.0099	0.0074	0.0099	0.0098	0.0099	0.0097	0.0094	0.0090
PenDigits	**0.0038**	0.0037	0.0037	0.0037	0.0037	0.0037	**0.0038**	0.0021	0.0021	0.0021	0.0021	0.0021	0.0021
Shuttle	**0.070**	0.0277	0.0274	0.0274	0.0266	0.02525	0.02347	0.0274	0.0272	0.0263	0.0261	0.0257	0.023
Waveform	0.0113	0.0381	0.0382	0.0379	0.0399	0.0385	0.0374	0.0338	0.0336	0.03371	0.0344	0.0397	**0.0410**
WBC	**0.3549**	0.3018	0.3016	0.3020	0.3032	0.3039	0.3021	0.2979	0.2977	0.2992	0.3001	0.2997	0.3021
WDBC	**0.2958**	0.2515	0.2513	0.2517	0.2527	0.2533	0.2518	0.2483	0.2481	0.2494	0.2501	0.2498	0.2518
WPBC	0.0708	0.0958	0.0958	0.0979	0.0959	0.1009	0.0930	0.1169	0.1169	0.1183	0.1203	0.1205	**0.1331**
Annthyroid	0.0073	0.0154	0.0155	0.0149	0.0144	0.0150	0.0133	0.0155	0.0157	0.0158	0.0160	0.0158	**0.0167**
Arrhythmia	**0.3047**	0.2542	0.2546	0.2559	0.2549	0.2513	0.2585	0.2428	0.2428	0.2453	0.2447	0.2474	0.2531
Cardiotocography	**0.1168**	0.0958	0.0958	0.0964	0.0968	0.0978	0.1012	0.0951	0.0951	0.0952	0.0955	0.0955	0.0955
HeartDisease	0.1969	0.1923	0.1932	0.1946	0.1941	0.1975	**0.1980**	0.1921	0.1929	0.1925	0.1923	0.1916	0.1938
Hepatitis	**0.2275**	0.2136	0.2141	0.2123	0.2141	0.2121	0.2161	0.2133	0.2129	0.2141	0.2154	0.2161	0.2167
InternetAds	**0.1539**	0.1262	0.1262	0.1271	0.1276	0.1289	0.1334	0.1253	0.1253	0.1254	0.1258	0.1258	0.1258
PageBlocks	0.1426	0.1407	0.1408	0.1411	0.1413	0.1415	0.1421	0.1394	0.1396	0.1397	0.1398	**0.1437**	0.1565
Parkinson	0.1864	0.2237	0.2278	0.2270	0.2296	0.2288	0.2213	0.2203	0.2223	0.2226	0.2247	**0.2319**	0.2303
Pima	0.0469	0.0500	0.0505	0.0520	0.0532	0.0545	0.0528	0.0528	0.0531	**0.0549**	0.0513	0.0534	0.0514
SpamBase	**0.0804**	0.05813	0.0581	0.0573	0.0588	0.0571	0.0572	0.0544	0.0544	0.0544	0.0545	0.0544	0.0543
Stamps	**0.1696**	0.1544	0.1551	0.1543	0.1548	0.1515	0.1534	0.1531	0.1528	0.1520	0.1527	0.1533	0.1519
Wilt	0.0066	0.0090	0.0089	**0.0091**	0.0090	0.0083	0.0062	0.0083	0.0082	0.0083	0.0081	0.0079	0.0075

Prediction Performance Analysis

Table 1.2 and Table 1.3 show the R@N and P@N results of our experiments. For clarity, all instances of $Ensemble_{AFS}$ are marked with * and all instances of $Ensemble_{UFS}$ with # in Table 1.2 and Table 1.3. The best performer for each dataset is highlighted in bold. The Friedman test illustrates that there is a statistically significant difference among the competitors for both R@N ($\chi^2 = 188.03$, $p < 0.001$) and P@N ($\chi^2 = 34.7$, $p < 0.001$). As such, we can safely reject the null hypothesis that all the algorithms perform the same. Once we verified that not all the performances of the algorithms are the same, the next step is analyzing which are different.

As such, we perform the Nemenyi test that compares all the one-class classification competitors to show statistically significant pairwise differences. For significance level $\alpha = 0.05$ we produce Fig. 1.3 and Fig. 1.4 that illustrate the *critical difference plots* of P@N and R@N respectively. In the *critical difference plots*, the classifiers that are not joined by a line can be regarded as different. In particular, Fig. 1.4 illustrates all instances achieve higher average ranking than the *Original baseline model* and that the great majority of the produced instances of our framework are not joined by a line with the *Original baseline model*. As a consequence, the great majority of our framework's instances achieve statistically significant greater R@N scores than the *Original baseline model*. On the other hand, the *critical difference plot* in Fig. 1.3 illustrates that there is no statistically significant pairwise difference between the instances of our framework and the *Original baseline model*.

Fig. 1.5 illustrates the P@N and R@N performance scores of all the competitors of our experimental setting. In green are coloured all the *UFS-RR* instances, in red all the *AFS-RR* and in black the *Original baseline model*. The perfect score is achieved when P@N and R@N are equal to 1.0; top right part of the plot. In addition, Fig. 1.5 shows the overall improvement of our framework on the 175 datasets that we used and is a visual representation of Tables 1.2 and 1.3. In Fig. 1.5, the vast majority of the datasets our instances hit right and higher parts of the plot than the *Original baseline model*. In cases where P@N scores are equal between our instances and the *Original baseline model*, greater R@N scores are observed for our instances.

Furthermore, Table 1.2 and Table 1.3 show that the instances of $Ensemble_{AFS}$ which are composed of more than 15 *AFS-RR* achieve the best P@N and R@N combined results on all the datasets. In particular, by taking into consideration both evaluation measures, P@N and R@N, the average combined ranking of *AFS-30RR* is 5.8, *AFS-25RR* is 6.2, *AFS-20RR* is 6.5 and *AFS-15RR* is 6.1 whereas the *Original baseline model* performs the worst among all the competitors; 13 in total. Any instance of the four aforementioned instances bring on average 18% statistically significant improved R@N and equal P@N scores on 175 datasets. Hence,

our extensive analysis shows that the augmented feature space does improve the novelty detection but deciding the best number of *AFS-RR* is non-trivial and possibly data-dependent. Taking into consideration the findings of our extensive analysis on a large variety of datasets characteristics, we argue that instances of $Ensemble_{AFS}$ which are composed of more than 15 *AFS-RR* is a safe choice.

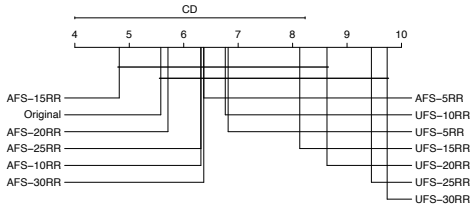

Fig. 1.3 Average ranking of all competitors over all data sets w.r.t. Precision@N; critical difference plot

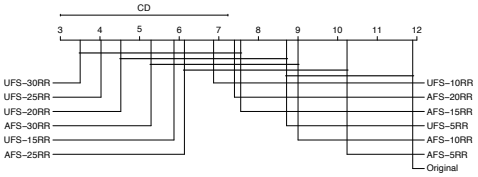

Fig. 1.4 Average ranking of all competitors over all data sets w.r.t. Recall@N; critical difference plot

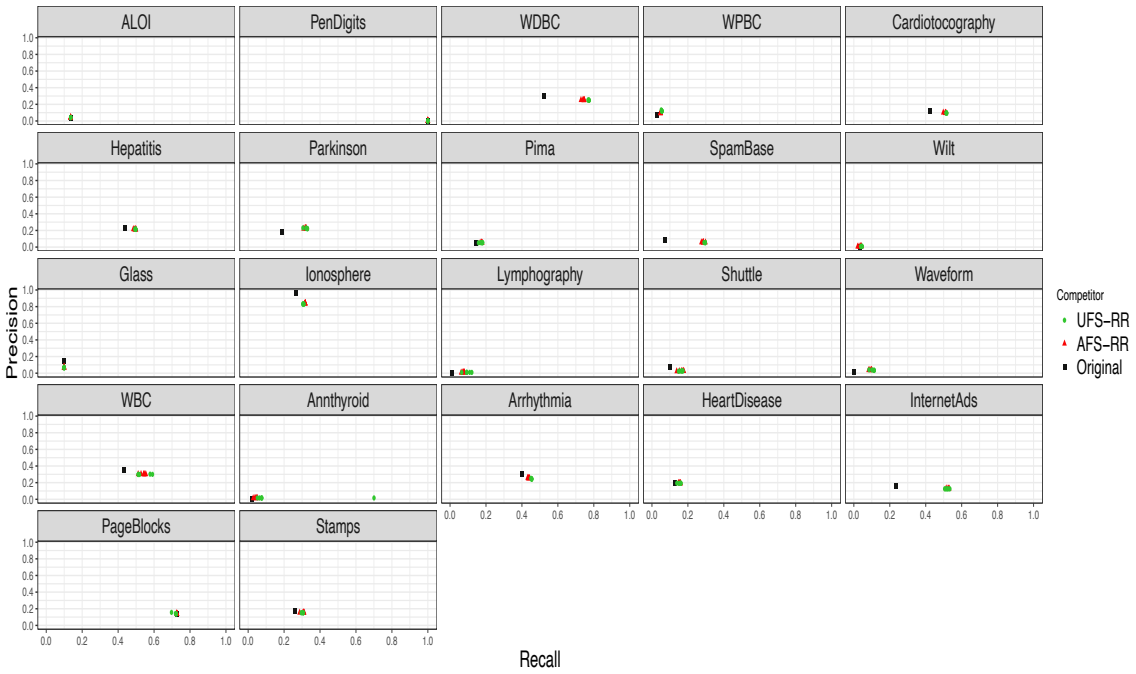

Fig. 1.5 Precision-Recall scores of all competitors on all datasets

Limitations and Future Directions:

This study produces our framework's instances in an unsupervised way. In other words, we do not leverage the ground truth to present the best performing instance but we investigate the performance of a finite number of instances. Recently, Campos et al. [44] and Rayana et al. [209] introduced unsupervised selection methods to find the best ensemble members.

As such, we plan to develop a framework that produces ensembles comprised of the best performing number of ensemble members. Furthermore, most of the existing OCC learners do not scale well [30] thus we prioritize a more scalable solution based on sampling techniques. Finally, explainable techniques are going to be incorporated to provide explanations to the user of our framework regarding the predicted novelties.

1.5 Conclusion

In this chapter, an one-class classification ensemble framework has been introduced to detect novelty. The proposed framework extends previous supervised outlier detection works [183, 184, 274] that used unsupervised scoring detectors to learn richer feature representations than the original feature space. However, in anomaly detection problems the most usual scenario is that sufficiently labelled training sets are absent and supervised approaches face considerable difficulties. In particular, there is in abundance training data that describe objects belonging to a particular class (usually normal objects) and for the fortunate scenarios there is a possibility to obtain very limited knowledge describing objects that do not belong to this class (abnormal objects). This work employs one-class classification algorithms which by design [242, 223] are able to address highly class imbalanced scenarios compared to fully supervised approaches that heavily depend on re-sampling (oversampling or downsampling) techniques. As such, a distinct direction is followed regarding how to accommodate such scenarios and in addition to address the inability of one-class learners to perform feature selection.

Our experimental results demonstrate statistically significant improvement in detecting novelties compared to baseline approaches. In particular, several OCC ensembles of the introduced framework bring on average 18% improved results. In our analysis, a benchmark data repository [45] is used to evaluate our proposed methodology. We select all the available downsampled dataset versions with the smallest rate of outliers to ensure that: (i) the presence of outliers is appropriate for the outlier detection task, (ii) our framework is well-tested in terms of robustness against different distributions of the outlier class. The introduced framework, analyzes 175 datasets and demonstrates enough robustness to handle a great diversity of scenarios in terms of number of attributes, percentage of outliers and the category they represent. To the best of our knowledge, this paper introduces for the first time a framework to detect novelties in a one-class classification setting.

Part IV

Conclusion and Future Work

Conclusion and Future Work

In this chapter, we recap the advancements and innovation of this doctoral thesis and we highlight the main contributions. Throughout the different chapters of this thesis we proposed advanced analytical techniques and novel machine learning methods to address real-life cyber and cyber-telecommunication problems. The current state-of-the-art is inadequate and fail to protect enterprises against adversaries. As such, we first proposed an ensemble method that requires very limited knowledge in order to detect the stealthy Lateral Movement attack. Then, we discussed unsupervised methods that are able to avoid the need of learning with supervision. More specifically, we developed two ensemble learning methods to address the problems of (i) detecting a stealthy and sophisticated cyber attack namely Lateral Movement attack, (ii) detecting fraudulent telephone calls made on the network of POST Luxembourg. We finally proposed an application domain agnostic method that learns unsupervised representations to improve novelty detection by developing one-class classification ensembles.

Supervised and Unsupervised Ensemble Learning to Detect the Lateral Movement Attack

The level of digitization in our modern age put enterprises in risk due to the increasing number of cyber threats. The situation escalates when adversaries form large teams, composed of more than one-hundred members, raise the level of sophistication. The Lateral Movement attack is one representative example of stealthy and well-organized attacks that targets big enterprises and organizations to mainly mine sensitive data. Such attacks are hard to be detected and usually it takes many days to be detected. We therefore proposed two methods based on supervised and unsupervised ensemble learning models to address the problem of detecting the Lateral Movement attack. Ensemble learning is a bunch of techniques that has demonstrated huge success especially in designing supervised learning methods that are based on either bagging or boosting to significantly improve the performance of

weak learners. Apart from supervised learning, ensemble learning is gaining popularity in the problem of anomaly or outlier detection and produces methods that are called outlier ensembles. As such, in this thesis we proposed two novel ensemble learning methods to address the real-life problem of detecting the lateral movement attack. Our experimental results are evaluated on the most popular public dataset related to the lateral movement and provided by the Los Alamos National Laboratory enterprise network.

Supervised Learning Approach

First, we proposed an independent and supervised outlier ensemble method to detect the lateral movement attack[1]. The components of this method are independently executed of one another. More precisely, we built our method by employing and combining predictions of three different classifiers, Random Forest, Logistic Regression and LogitBoost. The output of each individual model is a probability assigned to each data object that represents how likely is each data object to be an outlier. Majority Voting is used as ensemble learning technique that leverages the predictions of all individual models and gives the final prediction. In our experiments we underline that our proposed supervised ensemble is able to effectively detect the lateral movement attack despite very limited knowledge of the anomalous class. Additionally, our method enriches the initial set of features by proposing an advanced feature engineering strategy that is performed on graphs. We measured the performance of our method by using the false negative rate and false positive rate metrics. In overall we achieve 0 false negative rate (i.e. no attack was missed), and on average a false positive rate of 0.0019. In addition, the balanced accuracy metric is on average 99.62%.

Future Work

- In short term we aim to strengthen the graph-based feature engineering of our method. More precisely, we are going to represent authentication data as graphs and extract graph properties such as in- and out- degree of nodes, centrality indicators, and other metrics. In addition, we aim to employ stacking [262, 16] that is a well-performing ensemble machine algorithm. More specifically, stacking combines multiple models via a meta-learner to improve predictions.

- In long term we aim to integrate the evolution of bipartite graphs into the prediction process of the supervised learning ensemble. Temporal graphs are essentially graphs

[1]This chapter of the thesis was published in the 2018 NOMS IEEE/IFIP Network Operations and Management Symposium.

that change with time and gives us the opportunity to explore the dynamic properties of data. Moreover, our goal is to investigate active learning [6] methods for anomaly detection in order to give the opportunity to experts to guide the predictions.

Automatic Semi-supervised Learning Approach

Second, we proposed a sequential and unsupervised ensemble learning outlier ensemble method to detect the lateral movement attack [2]. The components of sequential ensembles are executed sequentially in such a way that there is a clear dependency between them. More precisely, we design our first component to be composed of an unsupervised ensemble using state-of-the-art unsupervised outlier detection algorithms. Regarding the second component of our sequential method we built an one-class classification ensemble that leverages the predictions of the first component. Additionally, the very first step of our method was to produce embeddings using the Logistic PCA technique, a variant of the popular PCA technique, to better represent the normal behavior. Our experiments showed that our proposed method that does not need supervision is able to improve the detection performance of the state-the-art methods. More precisely, our proposed detector outperforms existing algorithms and produces a 0 false negative rate without missing any malicious login event and a false positive rate which improves state-of-the-art approaches.

Future Work

- In short term, we are going to extend our work by improving the one-class classification ensemble. In particular, we are going to use multiple heterogeneous one-class classification algorithms instead of using one one-class classifier.

- In long term, we intend to replace the logistic PCA technique with network representation learning techniques and deep learning models in order to produce the embeddings. Additionally, we are going to investigate unsupervised feature selection methods to find subspaces that better expose anomalies.

[2]This chapter of the thesis was published in the 1st Workshop on Machine Learning for Cybersecurity (MLCS) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2019).

Unsupervised Ensemble Learning to Detect Telecommunication Fraud

The fraud ecosystem is relying more and more on the internet and fraudsters bring new types of fraud in order to get some unauthorized benefit. In particular, fraudsters can potentially target any technological system that involves money and services i.e. the credit card system and telecommunication system. In this chapter, we discussed the challenges and issues that any fraud detection systems have to deal with. We underlined the importance of knowing in advance the most common issues that a fraud detection system has to overcome in order to build effective detection systems. In addition, we extensively discussed and provided details related to different telephony fraud schemes and how researchers used data mining techniques to expose telephony fraud. We focused on a technology called Private Branch Exchange (PBX) which enables enterprise customers to manage their internal and external communication needs. Similarly to many other technologies PBX could be vulnerable to fraud activities in order to gain financial benefits. As such, proposed a machine learning method [3] that deals with the detection of fraudulent PBX phone calls made on the network of the largest provider in Luxembourg, POST Luxembourg. This method follows well-established unsupervised learning principles to ultimately build outlier ensembles. More specifically, several unsupervised outlier ensembles are developed to investigate the factors that affect the robustness of an outlier detection approach. Hence, an experimental research was conducted to highlight the impact of that factors on the performance of outlier ensembles in a real-world telecommunication fraud detection problem. The telecommunication area is the least studied area and due to multiple reasons there is a lack of academic work. This is also conformed by Abdallah et al. in their survey paper [4] that examines all published papers related to the most prominent fraud areas between 1994 and 2014. In this chapter, for the first time an unsupervised outlier ensemble method is developed for the problem of detecting fraudulent telecommunication activities. The experimental results of our method demonstrate the potentiality of similar approaches in the real-life fraud detection problem. More precisely, the isolation forest algorithm of our method managed to detect all fraud activities performed on POST's network by reaching an AUC score of 1.0.

[3] This chapter of the thesis was published in 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)

Future Work

- In short term we aim to enrich the feature space by learning unsupervised representations guided by outlier detection algorithms. Several one-class classification algorithms will be used to develop an ensemble and learn the normal behavior.
- In long term we aim to extend the work of Chen et al. [59] in unsupervised outlier ensembles using Autoencoders. In particular, we aim to develop outlier ensembles by investigating variants of the Autoencoders technique such as Variational Autoencoders, Adversarial Autoencoders and Generative Adversarial Networks.

One-Class Classification Ensemble Learning for Novelty Detection

Many real-life problems such as machine diagnostics, faults and failure detection in industrial systems, video surveillance, intrusion detection, and fraud detection aim to identify data objects that are not consistent with normal expectations. Our method focuses on this category of problems that are called novelty detection problems and typically include a training phase where the normal behavior is learnt. In addition, we shed light among the different terms that are used in the novelty detection context. Based on the literature in the anomaly detection field, one-class classification algorithms compose the most suitable approach in the novelty detection problem where their objective is to learn a decision function that distinguishes between normal and unusual observations. We proposed an innovative ensemble learning method that uses one the most popular one-class classification learners to address the problem of novelty detection, regardless the application domain of the problem [4]. More precisely, we took advantage of the merits of outlier scoring algorithms in order to learn multiple unsupervised representations. These unsupervised representations enrich the information included in the initial feature space by incorporating the concept of outlierness. In particular, outlier scoring algorithms such as LOF [39] assign an outlier score to each data object that represents the degree of *'outlierness'*. Our ensemble learning method relies on randomness that we induce to the way that unsupervised representations are learnt. As we discussed in Chapter 2 randomness is a fundamental element of ensemble learning and our method learns multiple random but informative representations in order to ultimately build an ensemble. In our extensive set of experiments, a benchmark data repository for outlier detection [45] was used. In particular, for our analysis we used 175 datasets that can be divided into two

[4] This chapter of the thesis is under submission

broad categories: 1. semantically meaningful datasets where the vast majority is related to medical applications, 2. datasets that have appeared in the outlier detection literature. Overall, our method managed to bring on average 18% statistically significant improved R@N (Recall@N) and equal P@N (Precision@N) scores on 175 datasets.

The diversity of the datasets in our analysis and the fact that 93% the datasets in our analysis contain less than 5% of outliers, ensures that our proposed method is more than well tested. We showed not only the ability of our method to effectively learn unsupervised representations but also the statistical significant improvement of our method compared to baseline approaches for novelty detection. Our method, extended existing works on supervised learning that leverage unsupervised representations. In particular, we successfully managed to move from fully-labeled scenarios to partially-labelled scenarios. In such scenarios, there is in abundance data that describe objects belonging to a particular class (usually normal objects) whereas there is very limited knowledge related to objects that do not belong to this class (abnormal objects).

Future Work

- In short term we aim to investigate techniques that are responsible for selecting the best performing ensemble members in an unsupervised way. An effective example of such techniques is the adoption of the boosting technique by the unsupervised learning field in order to produce better performing ensemble methods.

- In long we prioritize more scalable solutions because the majority of one-class classification learners do not scale well. Sampling techniques and deep learning methods are going to be investigated to overcome the scalability limitation.

References

[1] (2018). The mitre corporation. adversarial tactics, techniques & common knowledge. https://attack.mitre.org/wiki/Main_Page. Accessed: 2018-09-14.

[2] (2020). Remote desktop protocol. https://docs.microsoft.com/en-us/windows/win32/termserv/remote-desktop-protocol. Accessed: 2018-05-31.

[3] (2020). Sysmon v11.0. https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon. Accessed: 2020-04-28.

[4] Abdallah, A., Maarof, M. A., and Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113.

[5] Abe, N., Zadrozny, B., and Langford, J. (2004). An iterative method for multi-class cost-sensitive learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 3–11.

[6] Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509.

[7] Abe, S. (2005). *Support vector machines for pattern classification*, volume 2. Springer.

[8] Aggarwal, C. C. (2013). Outlier ensembles: position paper. *ACM SIGKDD Explorations Newsletter*, 14(2):49–58.

[9] Aggarwal, C. C. (2015). Outlier analysis. In *Data mining*, pages 237–263. Springer.

[10] Aggarwal, C. C. (2017). An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. Springer.

[11] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

[12] Aggarwal, C. C. and Sathe, S. (2015). Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter*, 17(1):24–47.

[13] Aggarwal, C. C. and Sathe, S. (2017). *Outlier ensembles: An introduction*. Springer.

[14] Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM.

[15] Akcay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2018). Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision*, pages 622–637. Springer.

[16] Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., and Vrahatis, M. N. (2019). Stacking strong ensembles of classifiers. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 545–556. Springer.

[17] Almeida, P., Jorge, M., Cortesão, L., Martins, F., Vieira, M., and Gomes, P. (2008). Supporting fraud analysis in mobile telecommunications using case-based reasoning. In *European Conference on Case-Based Reasoning*, pages 562–572. Springer.

[18] Alshawabkeh, M., Jang, B., and Kaeli, D. (2010). Accelerating the local outlier factor algorithm on a gpu for intrusion detection systems. In *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*, pages 104–110. ACM.

[19] Amrouche, F., Lagraa, S., Kaiafas, G., and State, R. (2019). Graph-based malicious login events investigation. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 63–66.

[20] Andrews, J. T., Morton, E. J., and Griffin, L. D. (2016). Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1):21.

[21] Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–27.

[22] Anthony, R. (2013). Detecting security incidents using windows workstation event logs. *SANS Institute, InfoSec Reading Room Paper*.

[23] Association, C. F. C. et al. (2013). Global fraud loss survey. *Press Release, Roseland, NJ (CFCA) October*, 10:2013.

[24] Aygun, R. C. and Yavuz, A. G. (2017). Network anomaly detection with stochastically improved autoencoder based models. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 193–198. IEEE.

[25] Badawi, E., Jourdan, G.-V., Bochmann, G., Onut, I.-V., and Flood, J. (2019). The "game hack" scam. In *International Conference on Web Engineering*, pages 280–295. Springer.

[26] Bai, Z. (2019). A machine learning approach for rdp-based lateral movement detection. Master's thesis, University of Waterloo.

[27] Ball, J. E., Anderson, D. T., and Chan, C. S. (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 11(4):042609.

[28] Barnett, V. (1978). Outliers in statistical data. Technical report.

[29] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

[30] Bengio, Y., LeCun, Y., et al. (2007). Scaling learning algorithms towards ai.

[31] Bian, H., Bai, T., Salahuddin, M. A., Limam, N., Abou Daya, A., and Boutaba, R. (2019). Host in danger? detecting network intrusions from authentication logs. In *2019 15th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE.

[32] Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

[33] Blanchard, G., Lee, G., and Scott, C. (2010). Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(Nov):2973–3009.

[34] Bohara, A., Noureddine, M. A., Fawaz, A., and Sanders, W. H. (2017). An unsupervised multi-detector approach for identifying malicious lateral movement. In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 224–233. IEEE.

[35] Borghesi, A., Bartolini, A., Lombardi, M., Milano, M., and Benini, L. (2019). Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9428–9433.

[36] Branco, P., Torgo, L., and Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50.

[37] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[38] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[39] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM SIGMOD Record*, volume 29, pages 93–104.

[40] Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20.

[41] Bullée, J.-W. H., Montoya, L., Pieters, W., Junger, M., and Hartel, P. (2018). On the anatomy of social engineering attacks—a literature-based dissection of successful attacks. *Journal of investigative psychology and offender profiling*, 15(1):20–45.

[42] Calvo, B. and Santafé Rodrigo, G. (2016). scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal, Vol. 8/1, Aug. 2016*.

[43] Campello, R. J., Moulavi, D., Zimek, A., and Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–51.

[44] Campos, G. O., Zimek, A., and Meira, W. (2018). An unsupervised boosting strategy for outlier detection ensembles. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 564–576.

[45] Campos, G. O., Zimek, A., Sander, J., Campello, R. J. G. B., Micenková, B., Schubert, E., Assent, I., and Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927.

[46] Canbek, G., Sagiroglu, S., Temizel, T. T., and Baykal, N. (2017). A comprehensive visualized roadmap to gain new insights. In *Computer Science and Engineering (UBMK), 2017 International Conference on*, pages 821–826. IEEE.

[47] Cao, H., Li, X.-L., Woon, Y.-K., and Ng, S.-K. (2011). Spo: Structure preserving oversampling for imbalanced time series classification. In *2011 IEEE 11th International Conference on Data Mining*, pages 1008–1013. IEEE.

[48] Cao, P., Zhao, D., and Zaiane, O. (2013). An optimized cost-sensitive svm for imbalanced data learning. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 280–292. Springer.

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

[50] Center, M. I. (2013). M-trends 2015: A view from the front lines. *Mandiant. com*.

[51] Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

[52] Chalapathy, R., Menon, A. K., and Chawla, S. (2018). Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*.

[53] Chan, P. K., Fan, W., Prodromidis, A. L., and Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, 14(6):67–74.

[54] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

[55] Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

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

[57] Chen, C., Liaw, A., Breiman, L., et al. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12):24.

[58] Chen, G., Zhang, X., Wang, Z. J., and Li, F. (2015). Robust support vector data description for outlier detection with noise or uncertain data. *Knowledge-Based Systems*, 90:129–137.

[59] Chen, J., Sathe, S., Aggarwal, C., and Turaga, D. (2017). Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 90–98. SIAM.

[60] Chen, K., Lu, B.-L., and Kwok, J. T. (2006). Efficient classification of multi-label and imbalanced data using min-max modular classifiers. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1770–1775. IEEE.

[61] Chen, M., Yao, Y., Liu, J., Jiang, B., Su, L., and Lu, Z. (2018). A novel approach for identifying lateral movement attacks based on network embedding. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 708–715. IEEE.

[62] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

[63] Chen, Y., Dang, X., Peng, H., and Bart, H. L. (2008). Outlier detection with the kernelized spatial depth function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):288–305.

[64] Chiang, A. and Yeh, Y.-R. (2015). Anomaly detection ensembles: In defense of the average. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 207–210. IEEE.

[65] Cisco (2018). What are the most common cyber attacks?

[66] Craswell, N. (2009). Precision at n.

[67] Daniel, T., Kurutach, T., and Tamar, A. (2019). Deep variational semi-supervised novelty detection. *arXiv preprint arXiv:1911.04971*.

[68] De Morsier, F., Tuia, D., Borgeaud, M., Gass, V., and Thiran, J.-P. (2013). Semi-supervised novelty detection using svm entire solution path. *IEEE transactions on geoscience and remote sensing*, 51(4):1939–1950.

[69] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

[70] Diehl, C. P. and Hampshire, J. B. (2002). Real-time object classification and novelty detection for collaborative video surveillance. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2620–2625. IEEE.

[71] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

[72] Ding, Z. and Fei, M. (2013). An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17.

[73] Dowle, M. and Srinivasan, A. (2017). data.table: Extension of 'data.frame'. R package version 1.10.5.

[74] Dowle, M. and Srinivasan, A. (2019). *data.table: Extension of 'data.frame'*. R package version 1.12.2.

[75] Duong, P., Nguyen, V., Dinh, M., Le, T., Tran, D., and Ma, W. (2015). Graph-based semi-supervised support vector data description for novelty detection. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

[76] Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd.

[77] Elmi, A. H., Ibrahim, S., and Sallehuddin, R. (2013). Detecting sim box fraud using neural network. In *IT Convergence and Security 2012*, pages 575–582. Springer.

[78] Emmott, A. F., Das, S., Dietterich, T., Fern, A., and Wong, W.-K. (2013). Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 16–21. ACM.

[79] Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36.

[80] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[81] Farvaresh, H. and Sepehri, M. M. (2011). A data mining framework for detecting subscription fraud in telecommunication. *Engineering Applications of Artificial Intelligence*, 24(1):182–194.

[82] FESTOR, O. (2017). *Understanding Telephony Fraud as an Essential Step to Better Fight It*. PhD thesis, TELECOM ParisTech.

[83] Fildes, N. (2018). Fraud costs telecoms industry $17bn a year.

[84] Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

[85] Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

[86] Frey, B. J. and Dueck, D. (2006). Mixture modeling by affinity propagation. In *Advances in neural information processing systems*, pages 379–386.

[87] Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

[88] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484.

[89] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.

[90] Gao, M., Hong, X., Chen, S., and Harris, C. J. (2011). A combined smote and pso based rbf classifier for two-class imbalanced problems. *Neurocomputing*, 74(17):3456–3466.

[91] Ghosh, J. and Acharya, A. (2011). Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305–315.

[92] Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

[93] Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220.

[94] Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

[95] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*.

[96] Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2013). Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262.

[97] Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

[98] Gruber, M., Schanes, C., Fankhauser, F., and Grechenig, T. (2013). Voice calls for free: How the black market establishes free phone calls—trapped and uncovered by a voip honeynet. In *2013 Eleventh Annual Conference on Privacy, Security and Trust*, pages 205–212. IEEE.

[99] Hagberg, A., Lemons, N., Kent, A., and Neil, J. (2014). Connected components and credential hopping in authentication graphs. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, pages 416–423. IEEE.

[100] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239.

[101] Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer.

[102] Hart, P. (1968). The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516.

[103] Hausknecht, M. and Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*.

[104] Hautamaki, V., Karkkainen, I., and Franti, P. (2004). Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 430–433.

[105] Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

[106] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE.

[107] He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

[108] He, H. and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

[109] Heard, N. and Rubin-Delanchy, P. (2016). Network-wide anomaly detection via the dirichlet process. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 220–224. IEEE.

[110] Henecka, W. and Roughan, M. (2014). Privacy-preserving fraud detection across multiple phone record databases. *IEEE Transactions on Dependable and Secure Computing*, 12(6):640–651.

[111] Hilas, C. S. and Mastorocostas, P. A. (2008). An application of supervised and unsupervised learning approaches to telecommunications fraud detection. *Knowledge-Based Systems*, 21(7):721–726.

[112] Hinde, S. (1996). Call record analysis. making life easier-network design and management tools (digest no: 1996/217). In *IEE Colloquium on*, volume 8, page 8.

[113] Hinneburg, A., Aggarwal, C. C., and Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *26th Internat. Conference on Very Large Databases*, pages 506–515.

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

[115] Hollmén, J. et al. (2000). *User profiling and classification for fraud detection in mobile communications networks*. Helsinki University of Technology.

[116] Holt, R., Aubrey, S., DeVille, A., Haight, W., Gary, T., and Wang, Q. (2019). Deep autoencoder neural networks for detecting lateral movement in computer networks. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 277–283. The Steering Committee of The World Congress in Computer Science, Computer ....

[117] Ighneiwa, I. and Mohamed, H. (2017). Bypass fraud detection: Artificial intelligence approach. *arXiv preprint arXiv:1711.04627*.

[118] Iranmanesh, S. A., Sengar, H., and Wang, H. (2012). A voice spam filter to clean subscribers' mailbox. In *International Conference on Security and Privacy in Communication Systems*, pages 349–367. Springer.

[119] Janssens, J. H. (2013). Outlier selection and one-class classification.

[Janssens and Postma] Janssens, J. H. and Postma, E. O. One-class classification with lof and loci: An empirical comparison. In *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, pages 56–64.

[121] Janssens, J. H. M., Flesch, I., and Postma, E. O. (2009). Outlier detection with one-class classifiers from ml and kdd. In *2009 International Conference on Machine Learning and Applications*, pages 147–153.

[122] Jansson, K. and von Solms, R. (2013). Phishing for phishing awareness. *Behaviour & information technology*, 32(6):584–593.

[123] Japkowicz, N. (1999). Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification.

[124] Jin, W., Tung, A. K., Han, J., and Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 577–593.

[125] Jolliffe, I. (2003). Principal component analysis. *Technometrics*, 45(3):276.

[126] Jungles, P., Simos, M., Godard, B., Bialek, J., Bucher, M., Waits, C., Peteroy, W., and Garnier, T. (2014). Mitigating pass-the-hash and other credential theft, version 2. *Microsoft TechNet*, page 60.

[127] Kaiafas, G., Varisteas, G., Lagraa, S., State, R., Nguyen, C. D., Ries, T., and Ourdane, M. (2018a). Detecting malicious authentication events trustfully. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE.

[128] Kaiafas, G., Varisteas, G., Lagraa, S., State, R., Nguyen, C. D., Ries, T., and Ourdane, M. (2018b). Detecting malicious authentication events trustfully. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE.

[129] Kang, P. and Cho, S. (2006). Eus svms: Ensemble of under-sampled svms for data imbalance problems. In *International Conference on Neural Information Processing*, pages 837–846. Springer.

[130] Karnouskos, S. (2011). Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pages 4490–4494. IEEE.

[131] Keller, F., Muller, E., and Bohm, K. (2012). Hics: High contrast subspaces for density-based outlier ranking. In *2012 IEEE 28th international conference on data engineering*, pages 1037–1048. IEEE.

[132] Kelty, C. (2011). The morris worm. *Limn*, 1(1).

[133] Kent, A. D. (2015a). Comprehensive, Multi-Source Cyber-Security Events. Los Alamos National Laboratory.

[134] Kent, A. D. (2015b). Cybersecurity data sources for dynamic network research. In *Dynamic Networks in Cybersecurity*. Imperial College Press.

[135] Kent, A. D. (2016). Cyber security data sources for dynamic network research. In *Dynamic Networks and Cyber-Security*, pages 37–65. World Scientific.

[136] Kent, A. D. and Liebrock, L. M. (2013). Differentiating user authentication graphs. In *2013 IEEE Security and Privacy Workshops*, pages 72–75. IEEE.

[137] Khan, S. S. and Madden, M. G. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374.

[138] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[139] Kiran, B. R., Thomas, D. M., and Parakkal, R. (2018). An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36.

[140] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., and Klein, M. (2002). *Logistic regression*. Springer.

[Kohavi et al.] Kohavi, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection.

[142] Kostopoulos, G., Livieris, I. E., Kotsiantis, S., and Tampakas, V. (2018). Cst-voting: A semi-supervised ensemble method for classification problems. *Journal of Intelligent & Fuzzy Systems*, 35(1):99–109.

[143] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

[144] Krebs, B. (2014). Target hackers broke in via hvac company. *Krebs on Security*.

[145] Kriegel, H.-P., Kröger, P., Schubert, E., and Zimek, A. (2009). Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1649–1652. ACM.

[146] Kriegel, H.-P., Kroger, P., Schubert, E., and Zimek, A. (2011). Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 13–24. SIAM.

[147] Kriegel, H.-P., Kröger, P., Schubert, E., and Zimek, A. (2012). Outlier detection in arbitrarily oriented subspaces. In *2012 IEEE 12th international conference on data mining*, pages 379–388. IEEE.

[148] Kriegel, H.-P., Schubert, M., and Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452.

[149] Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2-3):195–215.

[150] Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Citeseer.

[151] Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

[152] Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207.

[153] Kunene, G. (2019). A history of telecommunications: How telecoms became just another interface.

[154] Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., and Kim, K. J. (2017). A survey of deep learning-based network anomaly detection. *Cluster Computing*, pages 1–13.

[155] Landgraf, A. J. and Lee, Y. (2015). Dimensionality reduction for binary data through the projection of natural parameters. *arXiv preprint arXiv:1510.06112*.

[156] Last, F., Douzas, G., and Bacao, F. (2017). Oversampling for imbalanced learning based on k-means and smote. *arXiv preprint arXiv:1711.00837*.

[157] Latecki, L. J., Lazarevic, A., and Pokrajac, D. (2007). Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75.

[158] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 25–36. SIAM.

[159] Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166.

[160] Le, T., Tran, D., Tran, T., Nguyen, K., and Ma, W. (2013). Fuzzy entropy semi-supervised support vector data description. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5. IEEE.

[161] Lee, K., Kim, D.-W., Lee, K. H., and Lee, D. (2007). Density-induced support vector data description. *IEEE Transactions on Neural Networks*, 18(1):284–289.

[162] Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

[163] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

[164] Li, J., Huang, K.-Y., Jin, J., and Shi, J. (2008). A survey on statistical methods for health care fraud detection. *Health care management science*, 11(3):275–287.

[165] Lim, S. K., Loo, Y., Tran, N.-T., Cheung, N.-M., Roig, G., and Elovici, Y. (2018). Doping: Generative data augmentation for unsupervised anomaly detection with gan. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1122–1127. IEEE.

[166] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88.

[167] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008a). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE.

[168] Liu, X.-Y., Wu, J., and Zhou, Z.-H. (2008b). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550.

[169] Liu, Y., An, A., and Huang, X. (2006). Boosting prediction accuracy on imbalanced datasets with svm ensembles. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 107–118. Springer.

[170] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916. IEEE.

[171] Livieris, I. E., Kanavos, A., Tampakas, V., and Pintelas, P. (2018). An ensemble ssl algorithm for efficient chest x-ray image classification. *Journal of Imaging*, 4(7):95.

[172] Lopez, E. and Sartipi, K. (2018). Feature engineering in big data for detection of information systems misuse. In *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*, pages 145–156. IBM Corp.

[173] López, V., Fernández, A., García, S., Palade, V., and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250:113–141.

[174] Madsen, J. (2018). Ddoutlier: Distance and density-based outlier detection.

[175] Mani, I. and Zhang, I. (2003). knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126.

[176] Mann, M. I. (2012). *Hacking the human: social engineering techniques and security countermeasures*. Gower Publishing, Ltd.

[177] Marcos Alvarez, A., Yamada, M., Kimura, A., and Iwata, T. (2013). Clustering-based anomaly detection in multi-view data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1545–1548.

[178] Markou, M. and Singh, S. (2006). A neural network-based novelty detector for image sequence analysis. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1664–1677.

[179] Marques, H. O., Campello, R. J., Zimek, A., and Sander, J. (2015). On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, pages 1–12.

[180] Mathieu, B., Niccolini, S., and Sisalem, D. (2008). Sdrs: a voice-over-ip spam detection and reaction system. *IEEE Security & Privacy*, 6(6):52–59.

[181] McInnes, N., Wills, G., and Zaluska, E. (2019). Analysis of threats on a voip based pbx honeypot.

[182] Mehadi, A. (2018). *Performance Evaluation of Unsupervised Learning Techniques for Enterprise Toll Fraud Detection*. PhD thesis.

[183] Micenková, B., McWilliams, B., and Assent, I. (2014). 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*, pages 51–54.

[184] Micenková, B., McWilliams, B., and Assent, I. (2015). Learning representations for outlier detection on a budget. *arXiv preprint arXiv:1507.08104*.

[185] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[186] Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

[187] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[188] More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*.

[189] Moulavi, D., Jaskowiak, P. A., Campello, R. J., Zimek, A., and Sander, J. (2014). Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 839–847. SIAM.

[190] Müller, E., Schiffer, M., and Seidl, T. (2011). Statistical selection of relevant subspace projections for outlier ranking. In *2011 IEEE 27th international conference on data engineering*, pages 434–445. IEEE.

[191] Mygdalis, V., Iosifidis, A., Tefas, A., and Pitas, I. (2018). Semi-supervised subclass support vector data description for image and video classification. *Neurocomputing*, 278:51–61.

[192] Nguyen, V., Le, T., Pham, T., Dinh, M., and Le, T. H. (2014). Kernel-based semi-supervised learning for novelty detection. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 4129–4136. IEEE.

[193] Nicolau, M., McDermott, J., et al. (2016). A hybrid autoencoder and density estimation model for anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 717–726. Springer.

[194] Olszewski, D. (2011). Fraud detection in telecommunications using kullback-leibler divergence and latent dirichlet allocation. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 71–80. Springer.

[195] Orrite, C., Rodríguez, M., Martínez, F., and Fairhurst, M. (2008). Classifier ensemble generation for the majority vote rule. In *Iberoamerican Congress on Pattern Recognition*, pages 340–347. Springer.

[196] Papadimitriou, S., Kitagawa, H., Gibbons, P. B., and Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*, pages 315–326. IEEE.

[197] Parkinson, M. (1980). The extreme value method for estimating the variance of the rate of return. *Journal of business*, pages 61–65.

[198] Pasillas-Díaz, J. R. and Ratté, S. (2017). Bagged subspaces for unsupervised outlier detection. *Computational Intelligence*, 33(3):507–523.

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

[200] Pevnỳ, T. (2016). Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304.

[201] Phua, C., Lee, V., Smith, K., and Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

[202] Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215–249.

[203] Price-Williams, M., Heard, N., and Turcotte, M. (2017). Detecting periodic subsequences in cyber security data. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 84–90. IEEE.

[204] Pritom, M. M. A., Li, C., Chu, B., and Niu, X. (2017). A study on log analysis approaches using sandia dataset. In *26th ICCCN*, pages 1–6.

[205] Purvine, E., Johnson, J. R., and Lo, C. (2016). A graph-based impact metric for mitigating lateral movement cyber attacks. In *Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense*, pages 45–52.

[206] Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438.

[207] Rao, R. B., Krishnan, S., and Niculescu, R. S. (2006). Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter*, 8(1):3–10.

[208] Rayana, S. (2016). Odds library. *Stony Brook,-2016. NY: Stony Brook University, Department of Computer Science. URL: http://odds.cs.stonybrook.edu(2017)*.

[209] Rayana, S. and Akoglu, L. (2016). Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(4):42.

[210] Rendón, E., Abundez, I., Arizmendi, A., and Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5(1):27–34.

[211] Ritter, G. and Gallegos, M. T. (1997). Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18(6):525–539.

[212] Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., and Kloft, M. (2019). Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*.

[213] Ruff, L., Vandermeulen, R. A., Görnitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402.

[214] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[215] Sahin, M. and Francillon, A. (2016). Over-the-top bypass: Study of a recent telephony fraud. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1106–1117.

[216] Sahin, M., Francillon, A., Gupta, P., and Ahamad, M. (2017). Sok: Fraud in telephony networks. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 235–250. IEEE.

[217] Sahin, Y., Bulkan, S., and Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, 40(15):5916–5923.

[218] Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11.

[219] Sallehuddin, R., Ibrahim, S., Hussein Elmi, A., et al. (2014). Classification of sim box fraud detection using support vector machine and artificial neural network. *International Journal of Innovative Computing*, 4(2).

[220] Saravanan, P., Subramaniyaswamy, V., Sivaramakrishnan, N., Prakash, M., and Arunkumar, T. (2014). Data mining approach for subscription-fraud detection in telecommunication sector. *Contemporary Engineering Sciences*, 7(11):515–522.

[221] Sathe, S. and Aggarwal, C. C. (2016). Subspace outlier detection in linear time with randomized hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 459–468. IEEE.

[222] Sathe, S. and Aggarwal, C. C. (2018). Subspace histograms for outlier detection in linear time. *Knowledge and Information Systems*, 56(3):691–715.

[223] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

[224] Schubert, E., Wojdanowski, R., Zimek, A., and Kriegel, H.-P. (2012). On evaluation of outlier rankings and outlier scores. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 1047–1058.

[225] Schubert, E. and Zimek, A. (2019). ELKI: A large open-source library for data analysis–ELKI release 0.7.5 "Heidelberg". *arXiv preprint arXiv:1902.03616*.

[226] Schubert, E., Zimek, A., and Kriegel, H.-P. (2014a). Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 542–550.

[227] Schubert, E., Zimek, A., and Kriegel, H.-P. (2014b). Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237.

[228] Shi, L., Ma, X., Xi, L., Duan, Q., and Zhao, J. (2011). Rough set and ensemble learning based semi-supervised algorithm for text classification. *Expert Systems with Applications*, 38(5):6300–6306.

[229] Siadati, H. and Memon, N. (2017). Detecting structurally anomalous logins within enterprise networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1273–1284.

[230] Siadati, H., Saket, B., and Memon, N. (2016). Detecting malicious logins in enterprise networks using visualization. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE.

[231] Silver-Greenberg, J., Goldstein, M., and Perlroth, N. (2014). Jpmorgan chase hack affects 76 million households.

[232] Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, 19(5):631–645.

[233] Soria-Machado, M., Abolins, D., Boldea, C., and Socha, K. (2017). Detecting lateral movements in windows infrastructure.

[234] Sun, Y., Kamel, M. S., and Wang, Y. (2006). Boosting for learning multiple classes with imbalanced class distribution. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 592–602. IEEE.

[235] Sundarkumar, G. G., Ravi, V., and Siddeshwar, V. (2015). One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–7. IEEE.

[236] Swersky, L., Marques, H. O., Sander, J., Campello, R. J. G. B., and Zimek, A. (2016). On the evaluation of outlier detection and one-class classification methods. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10.

[237] Tang, J., Chen, Z., Fu, A. W.-C., and Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548.

[238] Tanha, J. et al. (2013). *Ensemble approaches to semi-supervised learning*. SIKS.

[239] Taniguchi, M., Haft, M., Hollmén, J., and Tresp, V. (1998). Fraud detection in communication networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 2, pages 1241–1244. IEEE.

[240] Tarassenko, L., Clifton, D. A., Bannister, P. R., King, S., and King, D. (2009). Novelty detection. *Encyclopedia of structural health monitoring*.

[241] Tax, D. (2005). Ddtools, the data description toolbox for matlab. *Delft University of Technology ed*.

[242] Tax, D. M. J. (2002). One-class classification: Concept learning in the absence of counter-examples.

[243] Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Machine learning*, 54(1):45–66.

[244] Teh, Y. W. (2010). Dirichlet process.

[245] Thai-Nghe, N., Gantner, Z., and Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. In *The 2010 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

[246] Thang, T. M. and Kim, J. (2011). The anomaly detection by using dbscan clustering with multiple parameters. In *2011 International Conference on Information Science and Applications*, pages 1–5. IEEE.

[247] Tomek, I. et al. (1976). Two modifications of cnn.

[248] Trittenbach, H. and Böhm, K. (2019). One-class active learning for outlier detection with multiple subspaces. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 811–820.

[249] Tsymbal, A. (2004). The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2):58.

[250] Tu, H., Doupé, A., Zhao, Z., and Ahn, G.-J. (2016). Sok: Everyone hates robocalls: A survey of techniques against telephone spam. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 320–338. IEEE.

[251] Turcotte, M., Moore, J., Heard, N., and McPhall, A. (2016). Poisson factorization for peer-based anomaly detection. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 208–210. IEEE.

[252] Turcotte, M. J., Kent, A. D., and Hash, C. (2017). Unified host and network data set.

[253] Ussath, M., Jaeger, D., Cheng, F., and Meinel, C. (2016). Advanced persistent threats: Behind the scenes. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 181–186. IEEE.

[254] Van Erp, M., Vuurpijl, L., and Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 195–200. IEEE.

[255] Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*.

[256] Verizon (2016). 2016 Data Breach Investigations Report. *Verizon Business Journal*, (1):1–65.

[257] Vilariño, F., Spyridonos, P., Vitrià, J., and Radeva, P. (2005). Experiments with svm and stratified sampling with an imbalanced problem: detection of intestinal contractions. In *International Conference on Pattern Recognition and Image Analysis*, pages 783–791. Springer.

[258] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19.

[259] Weiss, G. M. and Provost, F. (2001). The effect of class distribution on classifier learning: an empirical study.

[260] Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101.

[261] Williams, C. K., Engelhardt, A., Cooper, T., Mayer, Z., Ziem, A., Scrucca, L., Tang, Y., Candan, C., Hunt, T., and Kuhn, M. M. (2017). Package caret.

[262] Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

[263] Wu, G. and Chang, E. Y. (2004). Aligning boundary in kernel space for learning imbalanced dataset. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 265–272. IEEE.

[264] Wu, K., Zhang, K., Fan, W., Edwards, A., and Philip, S. Y. (2014). Rs-forest: A rapid density estimator for streaming anomaly detection. In *2014 IEEE International Conference on Data Mining*, pages 600–609. IEEE.

[265] Wu, Y.-S., Bagchi, S., Singh, N., and Wita, R. (2009). Spam detection in voice-over-ip calls through semi-supervised clustering. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 307–316. IEEE.

[266] Xing, D. and Girolami, M. (2007). Employing latent dirichlet allocation for fraud detection in telecommunications. *Pattern Recognition Letters*, 28(13):1727–1734.

[267] Yu, G., Zhang, G., Yu, Z., Domeniconi, C., You, J., and Han, G. (2012). Semi-supervised ensemble classification in subspaces. *Applied Soft Computing*, 12(5):1511–1522.

[268] Yu, J. (2015). Prevention of toll frauds against ip-pbx. In *Proceedings of the International Conference on Security and Management (SAM)*, page 259. The Steering Committee of The World Congress in Computer Science, Computer ....

[269] Yu, J. (2016). An empirical study of denial of service (dos) against voip. In *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, pages 54–60. IEEE.

[270] Yu, Y., Long, J., and Cai, Z. (2017). Network intrusion detection through stacking dilated convolutional autoencoders. *Security and Communication Networks*, 2017.

[271] Yu, Z., Zhang, Y., Chen, C. P., You, J., Wong, H.-S., Dai, D., Wu, S., and Zhang, J. (2018). Multiobjective semisupervised classifier ensemble. *IEEE transactions on cybernetics*, 49(6):2280–2293.

[272] Zhang, K., Hutter, M., and Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 813–822.

[273] Zhao, Y., Ding, X., Yang, J., and Bai, H. (2020). SUOD: Toward scalable unsupervised outlier detection.

[274] Zhao, Y. and Hryniewicki, M. K. (2018). Xgbod: improving supervised outlier detection with unsupervised representation learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

[275] Zhao, Y., Nasrullah, Z., Hryniewicki, M. K., and Li, Z. (2019a). Lscp: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 585–593. SIAM.

[276] Zhao, Y., Nasrullah, Z., and Li, Z. (2019b). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.

[277] Zhou, C. and Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674.

[278] Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

[279] Zimek, A., Campello, R. J. G. B., and Sander, J. (2014). Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):11–22.

[280] Zimek, A. and Filzmoser, P. (2018). There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6):e1280.

[281] Zimek, A., Gaudet, M., Campello, R. J., and Sander, J. (2013). Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 428–436.

[282] Zimek, A. and Schubert, E. (2017). Outlier detection. *Encyclopedia of Database Systems*, pages 1–5.

[283] Zimek, A., Schubert, E., and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387.