A Machine Learning Based Investigation of Cloud Service Attacks

By

Intisar Salem Hamed Al-Mandhari

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

April 2019

Abstract

In this thesis, the security challenges of cloud computing are investigated in the Infrastructure as a Service (IaaS) layer, as security is one of the major concerns related to Cloud services. As IaaS consists of different security terms, the research has been further narrowed down to focus on Network Layer Security. Review of existing research revealed that several types of attacks and threats can affect cloud security. Therefore, there is a need for intrusion defence implementations to protect cloud services. Intrusion Detection (ID) is one of the most effective solutions for reacting to cloud network attacks.

The application of machine learning-based techniques for the identification of network-based intrusion detections on the public dataset, KDD cup '99, has become commonplace since the last decade. This work has demonstrated that machine learning can be used to detect network intrusions. This thesis reports on an empirical investigation to determine the underlying causes of the reported poor performance of some well-known classifiers especially in learning from to classify minor classes/attacks. The investigations carried out in this thesis reveals that the KDD Cup '99 dataset is an imbalanced dataset due to the inherent nature of the network intrusion domain, where some attacks are very common and some attacks are rare. Therefore, there is an extreme imbalance among the number of data instances in the different attack classes of this dataset. Based on the number of the classes in the dataset, the imbalance dataset issue can be a binary-class problem or multi-class problem. In literature, most of the researchers focus on addressing binary-class misclassification problem as addressing the multi-class classification problem is a complex issue that requires detailed analysis of features of attacks and the performance and structure of classification algorithms used.

This thesis reports the basic methods that can be deployed to learn from an imbalance dataset. Different experiments are conducted to avoid the bias towards the major classes and enhance the detection rate of the classifiers used, especially in the classification of the minor classes. The findings show that the issue of learning from the imbalanced dataset is not due to the limitation of the classifiers but rather in the way they are structured and used in classification imbalanced datasets

Keywords: Cloud Computing, Intrusion Detection Systems, Network, Security, Attacks, Machine Learning, imbalance Dataset and Feature Selection.

Acknowledgements

First and foremost, praise is to Allah, the Almighty, on whom ultimately we depend for sustenance and guidance. Who was beside me all the time when no one was able to understand me and when the PhD stress reached its peak. My deepest gratitude is directed towards you, my lord, without your blessings, this achievement would not have been possible.

Special thanks to my soul mate who inspires me all the time, my darling husband, Mahmood Al-Azri, for his compromises, sacrifices, patience, understanding and endless support. To my soul, my little angel, my daughter, Durar, for providing me with both power and hope in life despite her small age. You are everything to me; you make me smile, happy and proud and from the moment they placed you in my arms, you have always snuggled right into my heart. I thank you both for putting up with me in difficult moments where I felt stumped and for goading me on to follow my dream to get this degree. Words cannot explain the depth of my gratitude towards you both.

I owe a lot to my parents, who encouraged and helped me at every stage of my personal and academic life, and longed to see this achievement come true. I deeply miss my father, who is not with me to share this happiness, who passed away in my third year of this journey. To the soul of my father, who taught me the endurance of success and life's meaning and who always believed in my ability to be successful. You are gone but your constant belief in my abilities has made this journey possible. To my mother, my paradise, the biggest source of my strength, the most precious gift I have in this life and to her prayers. Her dreams for me have resulted in this achievement and without her loving upbringing and nurturing; I would not have been where I am today and what I am today. It is true that if God ever existed, he would be in the form of a mother because only a mother can love and give without expecting anything in return. Had it not been for my mother's unflinching insistence and support, my dreams of excelling in education would have remained mere dreams. I thank my mother with all my heart and I know she is up there, listening, watching over me and sending me her blessings constantly and is my guardian angel.

To my brothers and my sisters, this would not have been possible without their unwavering and unselfish love and support is given to me at all times. To my mother in law, thanks for your prayers and caring. We represent a strong, big-hearted family with sturdy bonds; many thanks to you my lovely family and your prayers for me was what sustained me thus far.

I am a pleasure to sincerely thank my supervisor, Dr.Lin Guan who has given me her best guidance and support. Special thanks to my second supervisor, Pro. Eran A Edirisinghe for his continuous support, advice, help and invaluable suggestions throughout my PhD journey. His excellent guidance, constant motivation, steadfast encouragement and expert guidance make this journey a rewarding experience in my life that I will never forget. It is because of their utmost supervision that I have now achieved my own research accomplishments.

It would be inappropriate if I omit to mention the names of my dear friends; Alia, Sahar, Roqaya and Dania, who were there when I was away from my home and beloveds, who never let things, get dull or boring. Who have, in their own ways, kept me going on my path to success, assisting me as per their abilities, in whatever manner possible and for ensuring that good times keep flowing. To my friends from the power of the dream group, thank you for your ceaseless support and encouragement and for showing me the power of the dream. To those awaiting this accomplishment, I thank you for your prayers.

To all my teachers throughout my life who have participated in building my teaching and learning skills.

Last but not least, I would like to thank my sponsor, the Ministry of Higher Education, Oman, for providing me with this opportunity to complete my Ph.D. and for their financial assistance.

Publications

Accepted and Published Conference

- Al-Mandhari I.S., Guan L., Edirisinghe E.A. (2019) Investigating the Effective Use of Machine Learning Algorithms in Network Intruder Detection Systems. In: Arai K., Kapoor S., Bhatia R. (eds) Advances in Information and Communication Networks. FICC 2018. Advances in Intelligent Systems and Computing, vol 887. Springer, Cham
- I.S. Al-Mandhari, L. Guan, and E. A. Edirisinghe, "Impact of the Structure of Data Pre-processing Pipelines on the Performance of Classifiers when Applied to Imbalanced Network Intrusion Detection System Datasets," *in Advances in Information and Communication Networks*. IntelliSys 2019.

Glossary

- AFRL: Air Force Research Laboratory
- AI: Artificial Intelligent
- ANN: Artificial Neural Network
- APIs: Application Programming Interfaces
- ARP: Address Resolution Protocol
- BID: Behaviour-based Intrusion Detection
- BN: Bayes Net
- CIA: Confidentiality, Integrity and Availability
- CRISP-DM: Cross-Industry Process for Data Mining
- CFS: Correlation-based Feature Selection
- CSA: Cloud Security Alliance
- CSP: Cloud Solution Provider
- DARPA: Defence Advanced Research Projects Agency
- DDoS: Distributed DoS
- DIDSs: Distributed Intrusion Detection Systems
- DNS: Domain Name System
- DoS: Denial of Service
- DT: Decision Tree
- FNs: False Negatives
- FNR: False Negative Rate
- FPs: False Positives
- FPR: False Positive Rate
- FTP: File Transfer Protocol
- HIDSs: Host-based Intrusion Detection Systems
- IaaS: Infrastructure as a Service
- ICT: Information and Communication Technologies
- IDC: International Data Corporation
- IDSs: Intrusion Detection Systems
- IMAP: Internet Message Access Protocol
- IP: Internet Protocol
- KID: Knowledge-based Intrusion Detection

- KDD Cup 99: Knowledge Discovery and Data Mining
- MIT: Massachusetts Institute of Technology
- ML: Machine Learning
- MLP: Multi-Layer Perception
- NB: Navies Bayes
- NIDS: Network-based Intrusion Detection Systems
- NIST: National Institute of Standards and Technology
- PaaS: Platform as a Service
- PART: Partial Decision Tree Classifiers
- R2L: Remote to Local
- RF: Random Forest
- SaaS: Software as a Service
- SMOT: Synthetic Minority Oversampling Technique
- SVMs: Support Vector Machines
- TNs: True Negatives
- TNR: True Negative Rate
- TPs: True Positives
- TPR: True Positive Rate
- U2R: User to Root
- WM attack: WarezMaster attack
- WC attack: WarezClient attack
- Weka: Waikato Environment for Knowledge Analysis

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Research Background

Information technology is rapidly changing, and much of this is due to the introduction of Cloud Computing (CC) and the many applications supporting this technology. CC allows for resources to be distributed across a network, permitting users to interact with any of these resources as needed, with flexible access. As a result, convenience and security are increased, as users are not necessarily storing information independently. Companies have a greater ability to focus on the research and development of their own products but spend less time and energy designing secure data storage facilities, purchasing hardware for that purpose, or training staff to follow procedures if they are able to buy into cloud services offered by specialised cloud service providers. The many features and advantages of using cloud computing, including greater efficiency, lower costs, increased accessibility, reliability, and flexibility to manage and scale the systems make it very desirable to a variety of businesses and organisations in many different industries.

Unfortunately, cloud security is a significant concern for cloud users. As cloud usage relies heavily on the trust of users, there is a concern that organisations may be vulnerable to new threats and risks. Further, there is the possibility of cloud technology being invaded by intruders or hackers, thus giving them access to vital data in the cloud, owned by others. An intrusion or attack can have very significant implications for cloud usage. Since most attackers are likely to target networks with the largest numbers of users with the most accessible and most automated resources, as well as the networks containing an aggregate of the most information, it must be carefully investigated and all risks should be mitigated where ever possible. According to The Cloud Security Alliance (CSA), there are seven major threats to cloud systems [1]. Using Intrusion Detection (ID) methods is the best way to prevent attacks and defend the systems because these systems are able to rapidly recognise and therefore in a timely manner protect against attackers. Thus, it is imperative that any concerns surrounding cloud security be addressed before any benefits can be reaped.

1.2 Motivations

Because the internet is the delivery method of all cloud services, security is a priority [2]. Typically, any attacks incurred by a cloud system are unique to that cloud system. Recognising and eliminating the attacks is critical to maintaining the confidentiality and integrity of the cloud systems and the information and resources contained in those systems.

Outsider attacks by intruders are not the only threat to cloud security. While firewalls can successfully help prevent outside intruders from attacking a cloud system, only Intrusion Detection Systems (IDSs) will be helpful in detecting insider attacks. According to the study in [3], the costs required for adopting cryptographic strategies secure data are not always financially viable for attack prevention. Thus, using IDSs for cloud networks is an important consideration. Current IDSs are embedded with limitations, such as those related to the accuracy, sensitivity to false alarms, costs of communication, ideal detection rates, and coverage for attacks. Because of these limitations, many cloud systems are vulnerable to attacks and breaches of confidentiality. Finding solutions to these problems is critical to the integrity of any cloud system. Until solid solutions are identified, it will be difficult for consumers to fully trust cloud systems.

Research during the last decade has focused on developing different Machine Learning (ML) techniques for IDSs as mentioned in [4]. Strategies that are most commonly used are those that are able to learn from training samples illustrating typical network behaviours under different attacks. The IDSs strategically learn to detect intrusions, without the intervention of a human to identify the attack. The IDSs is able to recognise attacks after learning the typical patterns and variations seen during previous, known attacks.

One of the most popular datasets used to assess the performance of intrusion detection systems during the last ten years has been the KDD Cup 99 dataset (Knowledge Discovery and Data Mining) as mentioned in[4]. As there are many types of attacks, researchers apply different machine learning techniques to learn how to recognise such attacks. However in the previous work of using machine learning in IDS' researchers' have applied known classification algorithms for detecting various types of attacks. Majority of this work has been carried out by researchers who have the computer network security expertise, but not fundamental knowledge about machine learning algorithms. Therefore the attempts have only used the machine learning algorithm as a 'tool' to achieve detection of attacks, but many important aspects of the use and the essential fine-tuning needed in applying such tools have not been given the required importance. In particular, standard approaches to using machine learning classifiers to classify data within imbalanced datasets, often fail due to the fact that the number of training samples used to train minor attacks (i.e. attacks which are uncommon and hence not have enough samples to train the classifier model) is far less than the number of samples used to train to

detect major classes. For example attacks of type R2L (Remote to Local) and U2R (User to Root) are minor classes and their detection accuracies are often poor. The research presented in this thesis sets out to address this challenge.

1.3 Problem Statement

Effectiveness is the main factor considered when assessing the overall quality of an intrusion detection system. Effectiveness of an IDS is assessed by investigating its ability to detect intrusions accurately (true positive) against its rate of issuing false alarms (false positives). The review of literature conducted within the research context of this thesis revealed that there are three primary outstanding issues with regards to the machine learning-based IDS proposed in literature to-date.

Firstly, the literature fails to show consistency in the findings reported. Review of previously published research indicates that inconsistencies are likely to be derived from the researchers choosing to use different subsets of the KDD Cup '99 dataset in their research. Additionally, there are some methodological differences between how the data used for training has been pre-processed and the performance of the proposed machine learning systems have been validated.

Secondly, the popular dataset used for the research works, the KDD Cup '99 dataset, is not a balanced dataset. This becomes a significant issue when the number of instances in one specific class markedly outnumbers the number of instances in a different class, as this typically leads to a high number of misclassified instances for the class with fewer instances. Typically, the standard classification learning algorithms are often biased towards the majority class. One of the main problems with the KDD Cup '99 dataset being imbalanced is the issue that certain algorithms designed for machine learning (such as Decision Tree (DT) and Artificial Neural Network (ANN)) tend to be biased toward major classes of datasets (DoS and Probe[5]), which means that the minor classes (U2R and R2L[6]) end up with a poor rate of classification. However, the major classes and minor classes are clearly defined based on their proportion on the dataset, as presented in Table 3-1. The issue of an imbalanced dataset could be solved through balancing the data distributions in the dataset. The data imbalance can be due to an imbalance of instances between classes or imbalance of data instances within classes, with the latter arising due to the presence of the so-called sub-classes within classes. The data imbalance

between classes deals with the distribution of the instances between the dataset classes whereas data imbalance within classes deals with the data imbalance between sub-classes of a class.

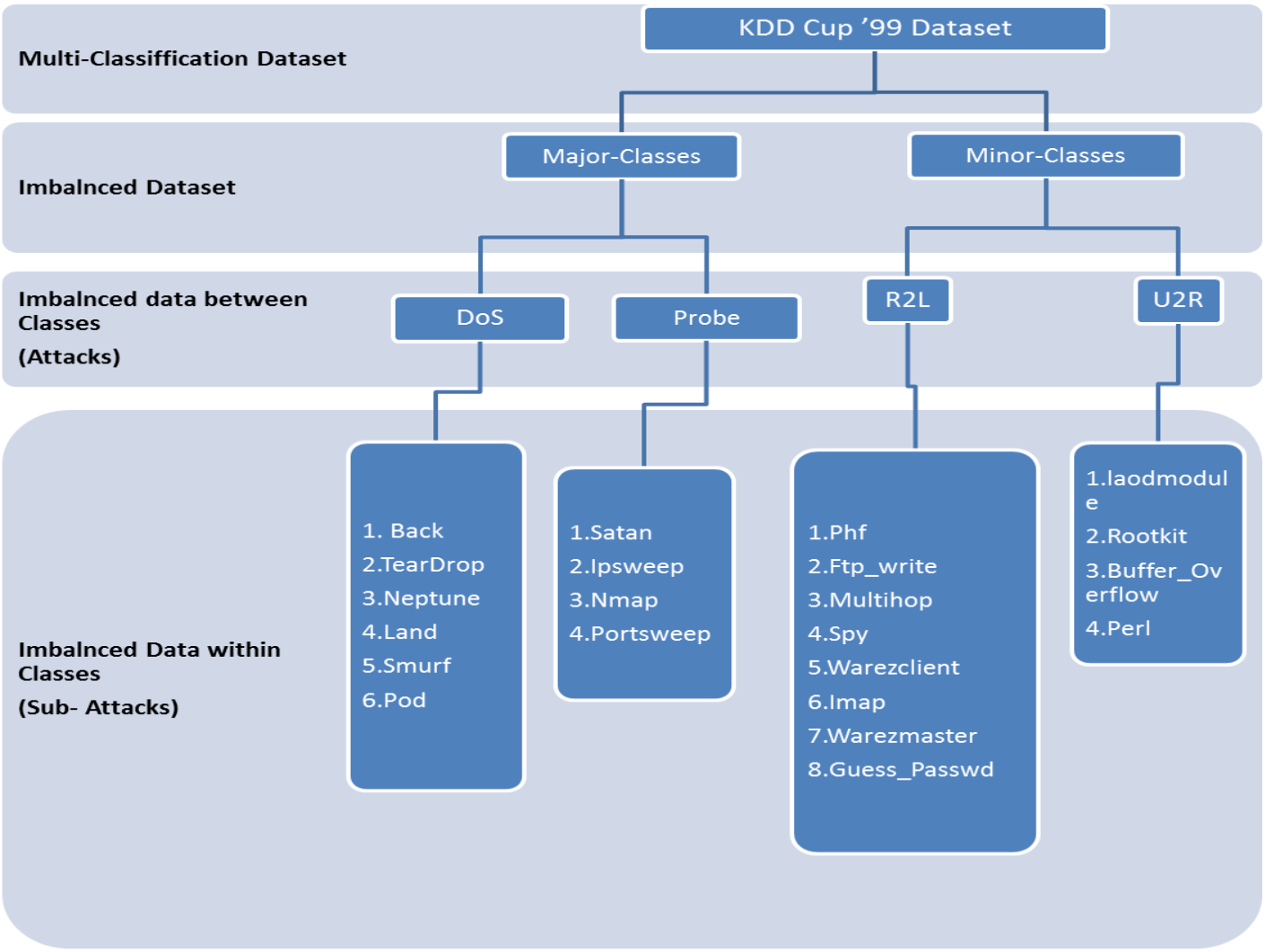


Figure 1-1:KDD cup'99 dataset imbalanced dataset issue

Figure 1-1 highlights the data imbalance issue within the KDD cup 99 dataset and the related presence of major/minor classes and sub-classes within them.

Thirdly, with a poor detection rate for minor attacks, even if the between class imbalances are addressed, it is required that assessment of minor attacks, as well as sub-minor attacks (attacks that are under the category of the minor attacks which can be defined based on their characteristics and where the KDD Cup '99 already classified it in their website), are carried out with care. Therefore this thesis will study the minor attacks and their sub-minor attacks. Our investigations revealed that sub-major classes do not affect the classification of major

classes as they have a sufficient number of instances to contribute to the training of a classifier. Further, each attack type has its own characteristic features and it is essential that these features are selectively used in a way that the classifier selection and performance can be optimised for the detection of both minor and sub-minor classes.

1.4 Aim and Objectives

This thesis aims to address outstanding research issues of present cloud security services (see Section 1.3) by offering an intelligent and tailored attack detection system based on machine learning algorithms and this will be satisfied by the following objectives:

Obj.1: Carry out a review of existing cloud security systems, to recognise the unique challenges, and explore potential solutions.

Obj.2: Investigate in detail current machine learning-based IDSs, to identify limitations in using standard machine learning approaches in attack detection.

Obj.3: Investigate the impact of data imbalance between classes and within classes on the performance accuracy of machine learning classifiers when applied to IDS.

Obj.4: Investigate in detail, the accuracy of detection of minor attacks and the impact of the presence of sub-minor classes.

Obj.5: Propose a novel machine learning framework that will address both between-class and within-class data imbalance.

Obj.6: Investigate the impact of feature-selection on classifier performance and how it can be used to optimise the performance of classifiers applied to an imbalanced dataset network detection attacks.

1.5 Methodology

A. First, an exhaustive review of the current cloud computing paradigm, architecture and research issues will be performed, leading to a deeper understanding of cloud system issues, challenges and shortcomings. This will lead to a deeper understanding of the research background required for this research. In particular, a comprehensive investigation about cloud security will be carried out leading to the understanding of security-related challenges and likely potential solutions.

B. Secondly, an investigation into the various IDSs strategies will be conducted alongside exploring the use of machine learning techniques for network intrusion detection.

C. Based on the approach followed in B, the strategies to improve the performance of classifiers in-network IDS will be examined. As a result, better-tailored learning methods will be applied based on classifiers, resulting in an improvement of detection accuracy of attacks.

D. The learning methods will be applied to the classification of attacks in a known imbalanced dataset, exploring the impact of both between class and within-class data imbalance on the classifier performance.

E. Given that feature selection is known to improve classifier performance in general, the thesis will explore the possibility of feature selection, depending on the type of attack, in order to classify attacks more accurately.

Figure 1-2 illustrates the scope of the research conducted with respect to the overall research context.

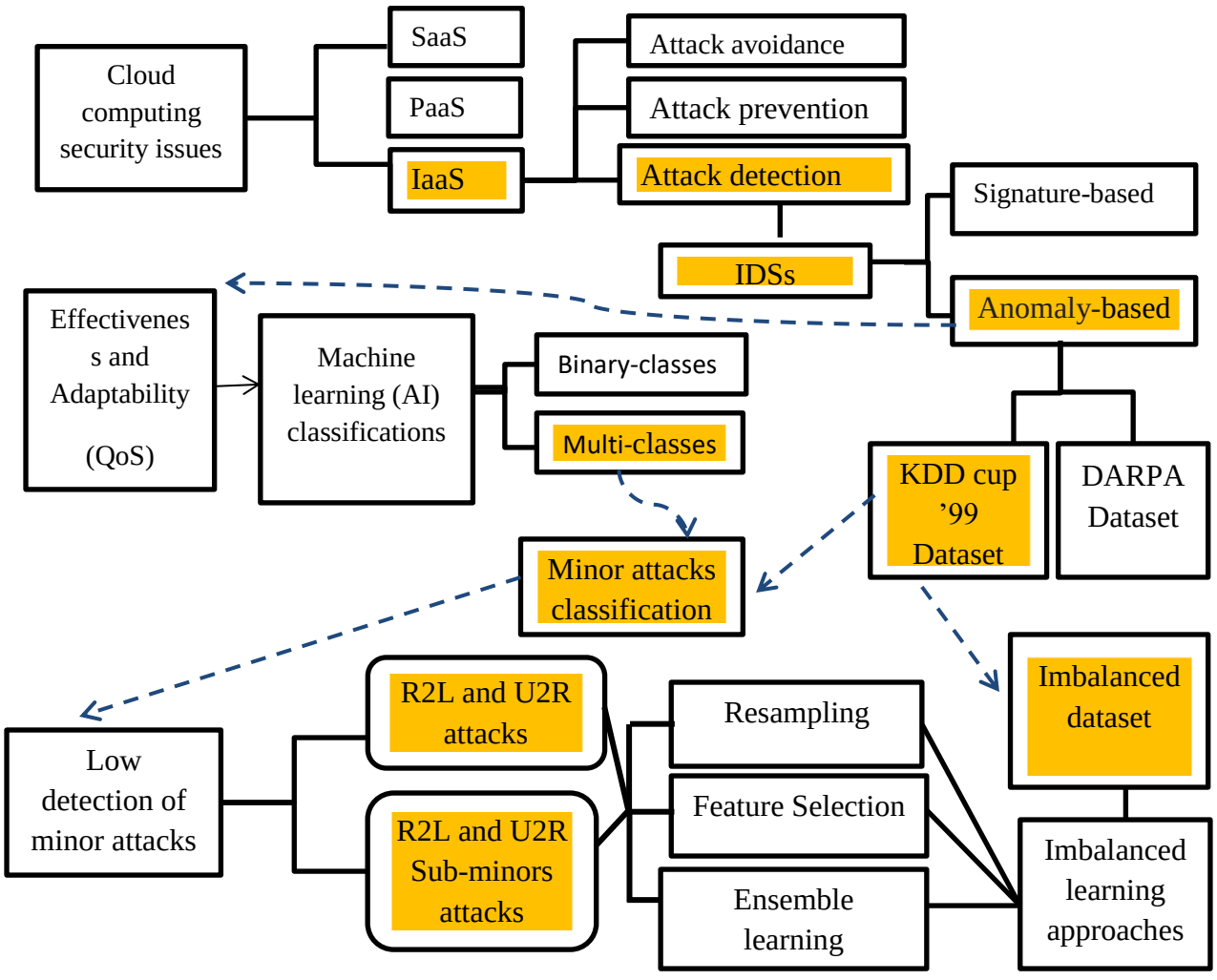


Figure 1-2:The scope of research investigation

1.6 Original Contributions

This thesis offers four contributions to the research area of network intrusion detection. While the focus of this research is mainly to contribute original knowledge to the area of intrusion detection systems, the proposed framework could also be used to deal with the classification of imbalanced datasets from other application domains. Hence the thesis also contributes to the general area of applications of machine learning algorithms.

1. Inconsistency & Poor Performance of Some Classifiers

The first contribution is a detailed investigation for the reasons behind inconsistency of the classification accuracy of machine learning algorithms reported in the literature for the purpose of network intruder detection.

- Regarding methods of validation, an experiment was conducted in this study which indicates that the result of the holdout validation differs based on the hold-out value that is deployed and the randomness of the training/test set selection. As a result, the tested instances may be different, depending upon the instances used in an alternative experiment, which may lead to an alternate detection rate. Based on this experiment, cross-validation was determined to be the ideal method, although it is computationally costly.
- The second achievement is related to the performance improvements obtainable from classifiers when resampling methods are used to pre-process data prior to training and testing. Based on this experiment, bias toward the majority classes were shown to be due to the data imbalance in the dataset used. After utilizing resampling techniques, all of the classifiers investigated perform well, improving accuracy significantly. The detection rate for minority classes (such as R2L and U2R) increased, and the bias toward the majority classes was less obvious. Balancing data between classes improved the accuracy of performance of all classifiers when dealing with imbalanced datasets. This was not reported in previous literature and is, therefore, a significant contribution of the research conducted within this thesis.
- The third contribution is related to the Random Forest classifier's behaviour in relation to adopting imbalance learning methods. This investigation led to the conclusion that resampling with ensemble methods, such as Bagging or AdaBoosM, is the best approach since this results in the RF classifier's rate of detection for minor classes improving without indicating a bias towards majority classes and poor classification of minor classes. It could be argued that RF performance is similar to the ensemble

method, however, it is determined that using performance of RF as an ensemble offers better information for classification than using it as a tree-based classification algorithm. This is due to the fact that RF utilizes subsets of features to split each node in a tree classification, and bagging (ensemble) methods incorporate all features required for node splitting. As a result, it can be concluded that the features in the dataset impact RF performance. Additionally, it is noted that RF is more effectively implemented with resampling, and if the ensemble method is used without resampling, then the RF detection rates are poor for minority classes.

- The fourth contribution is related to the Naive Bayes classifier and its ability to detect minor attacks. Although NB detection of the minority classes is improved by using a blended approach combining data-level methods with ensemble methods, the ideal strategy is using stacking (bagging of NB and RF).

2. The cause of poor detection of the minor attacks

The second achievement is derived from the study of reasons behind the poor performance of NB classifier in the classification of R2L attacks, despite using data resampling as a pre-processing stage. Due to the sub-classes of attacks, there are misclassifications. As certain features of R2L attacks are uniquely related to features of boundary/marginal attack classes, misclassifications may occur easily. As a result, the heterogeneous system being proposed is based on stacking and bagging, which improves the NB based detection of R2L attacks.

3. Impact of the Structure of Data Pre-processing Pipelines on the Performance of Classifiers

Based on the statistical analysis, both the sampling approach and the feature selection method should be used when any training for ideal classification methods is required. The strategy is based upon the classifier used and the type of features selected. The investigations in this thesis reveal that typically, the feature selection after resampling method will perform better than the other pipeline options.

4. Determine specific attributes of minor attacks and their sub-minor attacks and the significance of such features in the classification task

The research conducted in this thesis shows that feature selection algorithms can be effectively used to select the most prominent features that will in-turn be able to detect both minor and sub-minor classes more accurately when compared to using all possible features in

classification. This is a significant contribution as this not only demonstrates how the accuracy of classifiers can be improved via feature selection, but the feature selection also reveals vital information to the network IDS research community in understanding the key attributes of different kinds of attacks. This work is presented in Chapter-7

1.7 Thesis Outline

The thesis is organised into eight chapters. Chapter 2 introduces concepts in cloud computing security and presents a brief literature review, centred on examining the various attacks facing the Cloud systems, in addition to a summary pertaining to IDSs and its various categorisations. In chapter 2, the concept of machine-learning will undergo appraisal, with an investigation of adopting it to network intrusion identification. Chapter 3 presents the various challenges and considerations witnessed in regards to the classification of attacks when standard machine learning approaches are used without considering data imbalance issues present in typical network IDS datasets. Chapter 4 explores the adoption of various machine-learning algorithms in an effort to circumvent the usual misclassification issues experienced by academics that used the imbalanced KDD cup 99 datasets throughout the course of their research works. Chapter 5 investigates the performance of the popular Navies Bayes classifier in the presence of imbalanced data, specifically in classifying minor attacks, namely R2L and U2R. Chapter 6 investigates the use of data resampling to address the issue of class imbalance and how it should be combined with feature selection in a typical data pre-processing pipeline. The question of whether resampling should be performed before or after feature selection, for each classifier examined, is investigated. Chapter 7 carries out a rigorous investigation of the importance and ranking of features and their ability to positively attribute towards increased accuracy of attack detection is used as a reduced feature set. Finally, Chapter 8 concludes this thesis and suggests future improvements and enhancements.

Chapter 2 : Literature Review

2.1 Introduction

This chapter presents a literature review of Cloud Computing in general. Furthermore, it will seek to establish the security offered by Cloud Computing systems. Owing to the fact that the key objective is centred on improving Cloud-IDS security, there will be the presentation of a literature review, centred on examining the various attacks facing the Cloud, in addition to a summary pertaining to IDS and its various categorisations. Accordingly, the concept of machine-learning will undergo appraisal, with the specific adoption it to intrusion identification in cloud computing services.

2.2 Cloud Computing Overview

The most fundamental aspect of the Cloud can be identified in consideration to its component-centred nature, which is recognised as providing a number of different benefits, including customisability, extensibility, reusability, scalability and substitutability, the latter of which includes alternative adoptions, runtime component replacements and dedicated interfaces, all of which has been highlighted in the work of [7]. The key emphasis of Cloud Computing has been examined in the study of [7]which further established the key variances identifiable when comparing Grid Computing with the Cloud. Importantly, in the work written by [7], [8]the concept of Cloud Computing in the field of computer science was defined through the presentation of different definitions.

Nonetheless, Cloud Computing was defined by [9]as an IT implementation framework, centred on virtualisation, where there is the application of various applications, data and infrastructure-based resources through the internet, ‘as a distributed service by one or many different service providers’. As noted in the study by [9], such services are scalable on-demand and, furthermore, have their pricing structure determined in line with a pay-per-use approach. In this same regard, it is stated by [10]that Cloud computing utilises virtualisation technology in such a way so as to satisfy the objective to deliver computing resources as a valuable function. When comparing Autonomic Computing and Grid Computing with the Cloud, a number of different elements are recognised as comparable; nonetheless, there are also various aspects that differ between the three.

However, when considering the way in which Cloud Computing should be defined, the US National Institute of Standards and Technology (NIST) has provided a definition that has

undergone much development to become a de-facto standard. Accordingly, this particular definition is the one receiving the most support and is the most commonly cited, which highlights Cloud Computing as being a framework facilitating convenient, on-demand, universal access to a shared number of resources, whether applications, networks, servers, services or storage, for example, all of which is delivered and released in a time-effective manner without the need for any significant interaction or effort at the management of service provider side[11]. Importantly, the definition presented by NIST recognises a number of fundamental Cloud characteristics, as detailed as follows:

- Broad network access: Cloud resources can be accessed by the user through the network; this is facilitated by a variety of different platforms and mediums, including computers, mobile phones and tablets, for example.
- Measured service: There are the measurement and establishment of resource consumption owing to the fact that the user pays for what they use. This is provided through the application of a billing system.
- On-demand self-service: The service can be purchased and managed at the request of the user without the need for any human interaction.
- Rapid elasticity: The resources and services used by the consumer are the only ones actually paid for as the Cloud delivers a computing resources scalability function, which is able to be carried out dynamically.
- Resource pooling: The array of resources, spanning memory, network bandwidth, processing and storage, for example, are all brought together from the provider's computing resources, with different users sharing these. Importantly, the user is not given access to information pertaining to the location of the resources.

2.2.1 Cloud Computing Service Models

In consideration to the definition of Cloud Computing provided by NIST and also in line with CSA, Cloud Computing services are recognised as broken down into three fundamental services, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). In line with the service, there are a number of different control level variations, along with access between the providers and consumers of the Cloud.

SaaS provides software that is remotely accessible by consumers over the internet through the application of a usage-centred pricing framework. Through utilising SaaS, the provider's applications can be used by the consumer, which are notably run from a Cloud-based

infrastructure. Furthermore, the infrastructure, which is seen to encompass network, operating systems, servers and storage, are unable to be managed or controlled at the hands of the user. Nonetheless, the framework provided by SaaS is recognised as multi-tenant, meaning that a number of different users can make use of the architecture at the same time, but remains unique as far as the user's experience is concerned. In terms of the providers of SaaS, the most commonly utilised include Google Docs, Microsoft Office Online Services 365 and Salesforce.com.

Through PaaS, which is recognised as Platform as a Service, application developers are provided with an in-depth, wide-ranging development environment, with the CSP shouldering the responsibility for the configuration and installation of the virtual server, meaning this is not a task needing to be carried out by the user. Furthermore, the PaaS provider outlines the various program languages to be adopted and further provides APIs, databases, developer standards, libraries, toolkits, and software development environments, as well as payment and distribution avenues. Accordingly, this positions the user as being able to implement consumer-devised or gathered applications, generated through the use of various tools and programming languages, onto the Cloud infrastructure and supported by the provider. When it comes to the Cloud infrastructure forming the foundation of the system, the user has no control; however, the applications that are utilised is at the control of the user, and potentially so is the application-hosting environment configurations. Importantly, Google Apps, Force.com and Microsoft Windows Azure are all examples of widely used PaaS providers.

To this degree, Infrastructure as a Service (IaaS), which is essentially a somewhat generalised overview of the foundational hardware, and which is known to comprise mass storage systems, network components and PCs, is delivered. The provider of the Cloud ensures the Cloud is both available and usable, owing to the fact that the user is not able to gain access to its actual infrastructure. The customer in this regard is able to manage and make use of the resources available and is further positioned to utilise software, which might include applications and operating systems. Furthermore, the installation of additional services is also possible, with the user responsible for this, in addition to any connection established to an external system. Examples of commonly utilised IaaS providers include Amazon Web Services, Hosting.com and Rackspace.

2.3 Cloud Computing Security Concerns

In direct consideration to the environment and features offered by the Cloud, work in [12] presented the argument that Cloud Computing provides a significant number of additional security issues when compared with more conventional computing. On the other hand, others as in [13] consider Cloud security from the perspective of data protection, taking into account the fact that any organisation prioritises and highly values data as one of its key, most fundamental assets. Accordingly, the suggestion is presented in various works, namely those by [12], [13], that the Confidentiality, Integrity and Availability (CIA) of data needs to be guaranteed through the providers of such services; this will help to further facilitate the security of the Cloud. At the same time, confidentiality and availability are highlighted by[9] and identifying it as the Cloud's most fundamental security factors. Moreover, the range of restrictions and models between the consumers and providers of Clouds services result in weaknesses in the Cloud [14]. As has been established through the completion of the IDC(International Data Corporation) survey, owing to the presence of issues at privacy and security levels, in the work of [15] highlights that three-quarters of all users were found to have no inclination to move to Cloud Computing.

2.3.1 Cloud Attacks Classification

The point has been raised that an attack on the cloud has the potential to induce significant impact across both service and network, with media and network resources adopted by the attacker, which subsequently means a decline in the service performance, with the possibility that the network as a whole could collapse. It is acknowledged that there are three different types of attacks, namely interaction, penetration and mechanism, as highlighted in [16], [17] These are broken down as follows:

- **Interaction Type:** This particular group of attack is recognised through the interaction between the attacker and the network environment itself, with such attacks seen to be either passive or active in their nature. In the case of the former, such as through idle scan, port scanner or wiretapping, for example, a volume of important data is collected through tapping into traffic streams. In the case of the latter attack (active attacks), the attacker is known to influence the system resources operation or might otherwise opt to reconfigure them, such as through the adoption of ARP positioning, Denial of Service (DoS) attacks, Man-in-the-middle attacks, or Spoofing, for example. Importantly, such attacks are commonly problematic to identify owing to the fact that very little trace is left by the attacker.

- **Penetration Type:** In the case of either insider or outsider attacks, penetration is witnessed, with insiders seen to be authorised users utilising their own services in order to carry out illegal or otherwise harmful activities, or alternatively using other users' accounts. In the case of an outsider attack, the attack is initiated from beyond the borders of the network, with sensitive information garnered through scanning or probing attacks in an effort to subsequently implement actual attacks.

- **Mechanism Type:** In line with the various approaches and mechanisms adopted during initiation, it is possible to categorise the attack as belonging to one of the following: Denial of Service (DoS), Probing, Remote to Local, User to Root (U2R), and virus/worm attacks.
  - **A. Denial of Service (DoS) attack:** This particular attack has an impact on the availability of services through denying or otherwise restricting the access of users to the resources of a system, including, for example, bandwidth, buffers, memory and/or processing capability. When striving to make an attack successful, it is common for weaknesses in software to be positioned as the target, with changes made to the way in which a system is configured, and resources exploited to their limit. Such an attack might include ICMP Nukes, Land Attack, the Ping of Death, Teardrop, and changing a compromised router's configuration [18]–[20]
  - **B. Probe/Scanning attacks:** Through such attacks, networks are scoured for weaknesses or points of entry in order to gain access to network resources.
  - **C. Remote to Local (R2L) attack:** In this case, programs and commands encompassing local machine privileges are executed on the victim host after successfully circumventing the usual authentication process.
  - **D. User to Root (U2R) attack:** Higher level privilege is sought by attackers in achieving system access and control through achieving login access and accordingly circumventing the usual authentication process.
  - **E. Worm/virus:** Such an attack seeks to induce data loss, theft and dysfunction through the distribution of malicious code, which is implemented across a host or network.

Nonetheless, it is possible for attacks to be categorised in line with Cloud Computing surface attacks. As an example, a total of six different Cloud surface attack categorisations have been provided in the work of [21], as can be seen in Figure 2-1

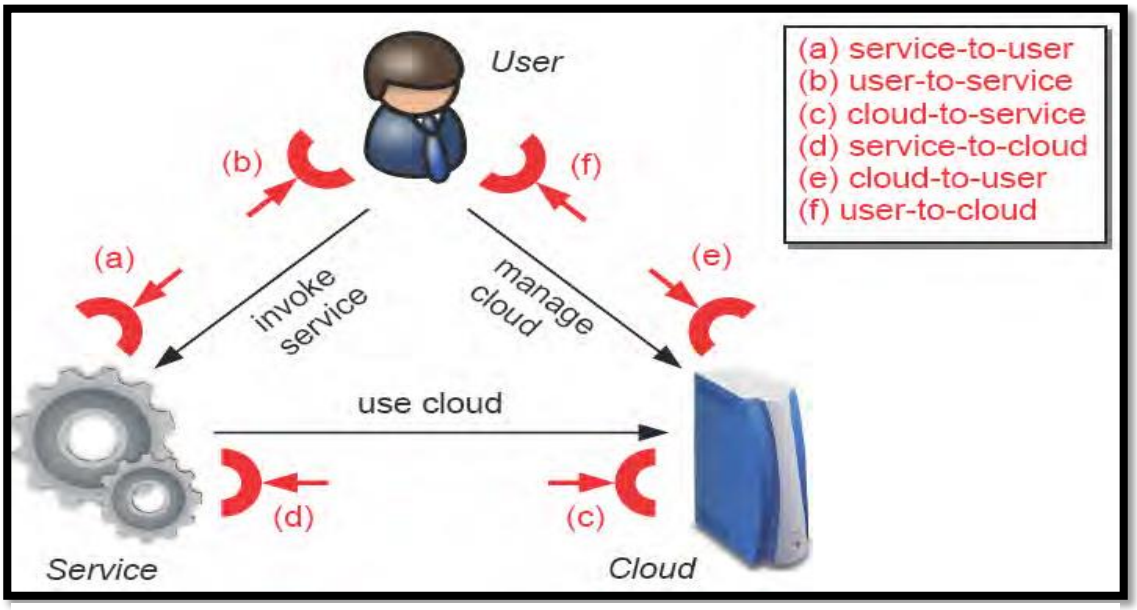


Figure 2-1: An attacks categorisation in regards cloud services [21]

Moreover, it is noted in the work of [22] that, in line with the attacker's behaviour in regards the weakness exploited or the type of mechanism utilised, attacks are defined as either host or network attacks:

- **Host-based attacks**: Such attacks may arise as a result of weaknesses in applications or operating systems. Buffer overflow, format string, and rootkit are a few examples in this regard.
- **Network-based attacks:** In this case, communications or interconnection structure confidentiality and integrity are attacked, with data modification, DoS attacks, eavesdropping, identity spoofing, IP address spoofing, and man-in-the-middle some examples of attacks on the network.

2.3.2 Intrusion Detection Systems (IDSs) Requirement in Cloud Computing

A number of attacks are critical and ultimately impact Cloud resources and services in terms of their availability, confidentiality and integrity. These attacks or intrusions can be a backdoor channel, flooding, insider, user root or virtual machine attacks. There is the suggestion that the Cloud's ultimate foundation is the network, meaning that any network weakness can ultimately impact the security of the Cloud as presented in [2], [23] Accordingly, network intrusion detection is recognised as amongst the most fundamental of security issues in Cloud

Computing, with network attacks adopting the form of DNS poisoning, DoS or DDoS attacks, insider attacks, IP spoofing, man-in-the-middle and port scanning as discussed in [2], [23]

When examining the most commonly utilised attack defences, these include IDSs and firewalls. Nonetheless, when it comes to managing the issues highlighted above, it is common for firewalls to be applied by the majority of Cloud providers, with a firewall able to protect the front access of the system, meaning the firewall is recognised as the first point of defence. Irrespective of firewall use, however, it is not possible for complex attacks or insider attacks, such as in the case of DoS or DDoS attacks, to be identified owing to the fact that only the network boundary's packets are sniffed. An organisation providing a Cybercrime Defender Platform referred to as Threat Metrix emphasises that 'You Can't Fight the Fire from Behind the Firewall'. With this noted, a Cloud-based platform cannot be protected by a traditional firewall owing to their inability to satisfy the Cloud's network security requirements. Accordingly, conventional firewalls are not recognised as a solid, effective approach to preventing all attacks [24]. One further approach to defence is through IDSs integration within the Cloud; this facilitates the monitoring and identification of internal and external network attacks across real-time network traffic. Moreover, a research was carried out in consideration to the security of Cloud Computing, with a survey implemented arriving at the conclusion that the identification of attacks and their prevention is recognised as the most pressing security consideration following the security of data.

2.4 An Overview of IDSs

2.4.1 Definitions and Terminology

The detection of attacks involves a process of identifying intrusion indicators in line with event supervision and accordingly examining what is seen to occur across ICT systems. Nonetheless, attacks are recognised as being a threat to or a violation of the security mechanisms or system components, where there is a compromise of availability, confidentiality and integrity. Attacks are implemented either inside or outside of the network, with inside attackers commonly exploiting the privileges they have been given and accordingly misusing them, whereas outside attackers commonly gain access to the system via the internet. Those systems that are seen to perform automated analysis and monitoring are referred to as Intrusion Detection Systems (IDSs). Moreover, IDSs is implemented by network and system administrators for a number of reasons, as detailed as follows in consideration to the work of [25]:

- To help ensure illegal behaviour occurrence is averted via improving the levels of risk regarding identification and penalty.
- To facilitate security measures through further encompassing the attacks and security infringements not averted by it.
- In mind of the identification and management of attacks.
- Advising the entity in regards to the presence of risks through documentation.
- Providing a high level of service quality, particularly in the case of complex enterprises, through ensuring the security administration and design quality is controlled.
- To enhance the identification, improvement and correction of root causes through the deliverance of valuable data pertaining to the attacks that have been identified.

As has been highlighted in the studies of [26]–[28] availability, confidentiality and integrity are recognised as being amongst the most prominent and pressing of security metrics for application when it comes to system quality assessment. In order to ensure such approaches, there is a need for IDSs to work both accurately and efficiently in the identification of instances of intrusion. Nonetheless, as noted in the works of [29], [30]the overall efficiency of IDSs is determined through the following:

- **Accuracy and Precision:** When there is inaccuracy in IDSs flags, this is recognised as identifying genuine actions as anomalous or intrusive. In actual fact, in regards to the outcomes of IDSs, there are four potential possibilities, as determined in Table 2-1. These are True Negatives (TNs) and True Positives (TPs) (which are seen to be aligned to a particular IDSs operation upon the successful labelling of events as either 'normal' or 'attack'); or alternatively, False Positives (FPs) (in relation to false alerts provided by IDS, where a normal event is labelled as a potential attack) and False Negatives (FNs) (false alerts, which arise upon the occurrence of attacks, which are erroneously referred to as normal events as discussed in [31], [32]

Table 2-1: Possible statuses for an IDS reaction

		Predicted	
		Normal	**Attack**
Actual	**Normal**	True Negative(TN)	False Negative(FN)
	Attack	False Positive(FP)	True Positive(TP)

When seeking to measure the overall precision of IDSs, the rate of the reaction detailed above is calculated through the application of the following as mentioned in[33]:

$$True\ Negative\ Rate(TNR) = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = \frac{\mathrm{No.\ of\ true\ alerts}}{\mathrm{No.\ of\ alerts}}$$

$$True\ Positive\ Rate(TPR) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{\mathrm{No.\ of\ detected\ attacks}}{\mathrm{No.\ of\ observable\ attacks}}$$

$$False\ \ Positive\ Rate(FPR) = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}$$

$$False\ Negative\ Rate(FNR) = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}$$

$$Accuracy\ = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

$$Precision\ = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Importantly, accuracy is seen to relate to the percentage of attacks that have been seen to arise and which have accurately been identified by the IDSs.

- **Performance**: The performance of IDSs relates to the rate at which audit events are processed through IDSs. Should the identification be carried out not in real-time, the performance of the IDSs would then be assessed as inadequate.

- **Completeness:** Those IDSs that are viewed as incomplete are unsuccessful in attacks identification. Nonetheless, as a result of the shortage of in-depth insight pertaining to attacks, such an approach is complicated when it comes to assessment.

2.4.2 The Classification of IDSs

2.4.2.1 Classification Based on Type of Data

In line with information sources and projected objectivities, IDSs can be categorised into three key classifications as mentioned in the work of[34]:

- **Network-based intrusion detection systems (NIDSs)**

The key commercial type of IDS is network-based referred to as NIDSs, which gather and examine network packets with the aim of identifying attacks. Furthermore, this type of IDS is able to monitor network traffic across different hosts linked to the segment through performing listening on the network segment or switch. It is common for NIDSs to contain a number of different single-purpose hosts or sensors at a number of different locations across a network. Accordingly, attack reports are communicated from such units following the monitoring of network traffic, with the central management console receiving this data, with a local traffic examination then carried out. In consideration of sensor security, this is simply achieved through the fact that the sensors are restricted to IDSs operation. Importantly, at the present time, sensors are designed in such a way so as to operate in what is referred to as 'stealth' mode, which therefore requires that a number of problems are experienced by attackers in establishing the correct location, or even just their presence [25]. In addition, there has been the suggestion that breaking down a network into individual parts, notably through the application of switches, is recognised as the most valuable approach when seeking to achieve large-scale network security. Accordingly, the individual parts of the network are well secured through the use of security technology, including IDSs and firewalls, for example as suggested in the work of[35].

In actual fact, when examining the benefits associated with the use of NIDSs, it is important to highlight that they are able to be made invisible, which therefore means the attacker is unable to find them. In addition, owing to the fact that NIDSs are recognised as passive instruments, the application of such does not have any effect with regards to existing networks. A well-sized network is successfully observed with the use of a handful of NIDSs should they be positioned in beneficial spots [25]. On the other hand, however, drawbacks may still be present, including the fact that, when the network is busy or have a notable size, it is not always possible to complete packets analysis, meaning that all of the attacks arising throughout a time of high traffic might not be successfully identified. One further drawback to the use of NIDSs is seen when considering that not every advantage can be garnered in the case of modern-day switch-

based networks: although switches implement the breaking down of networks into different parts, it remains that the majority of switches cannot deliver universal port monitoring. Accordingly, NIDSs sensors' monitoring ranges are somewhat restricted. Furthermore, it is not always possible for NIDSs to complete analysis on encrypted data and, with the increased adoption of virtual private networks by companies, the issue is becoming greater. Moreover, when there is an attack attempted, the user becomes informed of this, despite not being able to determine whether or not the attack has been a success. As such, if administrators are to establish this, they then have to complete manual investigations on the attacked host. As a further drawback, NIDSs are recognised as lacking stability and can, therefore, crash in those instances where segmented packets are involved in network-based attacks.

- **Host-based intrusion detection systems**

An IDS that is seen to be host-based exists on the endpoints of a network owing to the fact that it is positioned on a host so as to facilitate its gathering and supervision of suspect data and events witnessed across it. Due to the fact that the design of HIDSs is carried out in order to facilitate functionality on particular hosts, such as in the case of web servers or mail, for example, HIDSs demonstrate a significant degree of precision and reliability. Furthermore, any attack and the way in which they impact processes and users can also be established, as highlighted in the work of [35].

As an additional ability offered by HIDSs, they are able to complete examination on encrypted packets, and therefore demonstrate sound functionality across switched networks, which therefore position them as a valuable complement to NIDSs. Furthermore, they have the ability to identify attacks that otherwise would not be identified by NIDSs, predominantly owing to their supervision of host-local events; HIDSs are unable to identify attacks beyond their borders. Nonetheless, the management of HIDSs is problematic, as can be seen when considering that all monitoring hosts' information requires management and configuration as mentioned in the work of [36].

- **Distributed-based intrusion detection systems**

In the case of DIDS (distributed IDS), a number of different IDSs (such as HIDS and NIDS, for example) are seen to be encompassed across a large network. This type of IDS communicates across a hierarchical architecture with a number of different servers or otherwise a central service, which completes monitoring of the network. As can be seen in the Figure 2-

2, the IDSs' hierarchical tree-like structure can be observed, where circles are representative of network nodes and the arrows represent the flow of data across various node types. The points of gathering to host- or network-based systems can be seen represented by the leaf nodes. Importantly, data from a number of different nodes are aggregated by the internal nodes, meaning the leaf node gathers the data that is communicated to the internal nodes. Subsequently, at the greatest node level, there is the operation of additional aggregation, abstraction and reduction of data until the root node is reached. Following, attack signatures are then assessed by the root node, with responses issued and a report communicated to an operator console, which is regarded as being the control and command system. Importantly, however, it is the administrator that carries out the role of assessing issue and status commands. It is posited that such a structure means direct attacks could be a weakness of IDSs.

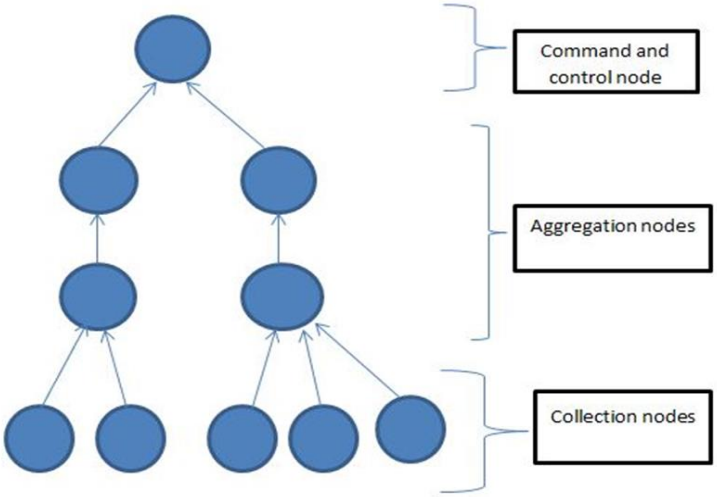


Figure 2-2: Distributed IDS

In addition, in instances where IDSs are lacking communication lines or, at a minimum, the capacity to complete dynamic relationship configuration in the event of component failure, there is the potential of a number of different failure points. As has been highlighted in the study by[21], owing to the fact that survivability approaches, including dynamic recover, mobility and redundancy are not widely applied in the case of presently implemented IDSs, weaknesses may still be present. As has been highlighted by[29], EMERALD is one example of a DIDSs, whilst INBOUNDS is recognised by[37].

2.4.2.2 Classification Based on Detection Approaches

As has been presented a study by[38], there are two key methods to be adopted in the identification of attacks, namely KID (Knowledge-based Intrusion Detection) and BID

(Behaviour-based Intrusion Detection). In the case of the former, attack evidence is sought in line with the data gathered that is seen to align with previously established attacks. It should be highlighted that KID is also referred to as misuse intrusion detection, rules-based intrusion detection, and signature-based intrusion detection. At the same time, through the latter method, BID, key deviations away from activities that are recognised as normal in line with the observations of the system throughout the phase of normal status are sought. BID is also commonly referred to as anomaly-based intrusion detection.

A. Knowledge-based intrusion detection

KID, fundamentally, relies on prior knowledge of vulnerable aspects within the system, how these could be exploited, and previous attacks that have taken place. These three components of information are held within a knowledge-based intrusion detection system (KIDS), which can objectively analyse incoming data, focusing on these specific components. Events that are not distinctly correlated to this stored information will be accepted, as the IDSs will not recognise it as an attack. When a behaviour that matches a previous attack is detected, it is flagged, ensuring quick recognition of an attack. In order for such knowledge-based systems to operate effectively, it is essential for the system to be updated in a timely manner with accurate information about previous attacks. Provided the information is accurate, there is a limited risk of type one, false-positive errors, and a high propensity for attack detection.

Whilst this is a key asset, the system relies solely on attacks being replicated. If an attack takes on a new form, even with a relatively subtle change, the system will not detect it. The system is only capable of learning from the experience of previous attacks and therefore cannot protect against new intrusions. There are a number of KIDSs available, with the most popular being Snort, which uses signature analysis, expert systems and state transition analysis to detect attacks as discussed in [22], [25]

1. **Expert Systems:** Within these systems, any attack is explained using a specific set of criteria included within IDSs in general. With expert systems, any event that is reported is first translated into a specific list of facts that define their meaning within the expert system. This list of criteria and facts is translated into rules that are subsequently processed by the KID to draw conclusions. Systems using the expert system approach must systematically browse the trail of incidents to determine patterns and assign

meaning to the data and facts. This approach is considered to require a rule-based language to generate information about attacks.

2. **Signature Analysis:** Similarly to expert systems, this technological approach also requires the gathering of knowledge. Although, the way in which the information and knowledge are exploited differs. The attacks are classified according to the semantic level, yet this approach decreases the description required to be stored on any audit trail. Many commercial level IDSs utilize this approach because it is considered to be highly effective.
3. **State transition analysis:** Unix was the first system to utilize this technological approach. After being successfully applied in Unix, many other systems began to use this method. The state transition analysis method uses groups of goals and transitions to describe attacks. The attacks are then diagrammed according to various characteristics. This method is often compared to model-based reasoning methods.

B. Behaviour-based intrusion detection

This method of intrusion detection is based on ongoing observations of normal activity in a system. Once patterns of normal activity are established, any abnormal behaviour or deviation from the routine behaviours may signal an intrusion. Every IDS keeps track of the normal patterns of all users, including the network connections, hosting information, applications accessed, and more. This information is collected and developed into a user profile, or a model for comparison. The profile is comprised of a series of activities collected and updated over a certain period of time. Thus, the IDS will report an attack any time there is a significant deviation when current patterns and stored patterns of activity are compared. Behaviour-based intrusion detection, therefore, is able to detect both known and unknown attacks, and may even be able to detect any new attacks on a system. Attacks do not have to specifically exploit any security weaknesses or vulnerabilities to trigger discovery by BIDs. This means that this type of detection system can detect attacks in the form of abuse of privileges.

Unfortunately, there is a high rate of false alarms in BID systems, which is a significant drawback. False alarms can raise issues with behavioural patterns. Additionally, there is required retraining over time as behaviours may not remain entirely stable by users. Without this retraining, more false alarms will occur. Behaviour-based intrusion detection systems, or

BBIDSs, tend to use four different approaches to detect intrusion, statistics, expert systems, neural networks, and user intention identification.

1. **Statistics:** According to the idea in[39], statistics is the most widely used approach for developing behaviour-based IDSs. This is because there are a wide variety of samples that can be measured across time, including login and logout times, number of resources per session, and duration of resources. Samples can be generated over time and can take a few minutes or more than a month to collect full samples. Averages for each variable are calculated and applied to the original model. Using standard deviations, thresholds are identified that, when exceeded, help detect attacks.
2. **Expert Systems:** Although this approach is considered to be less efficient than using statistics because it cannot provide as much data and information to be audited, it can still be a useful way to collect information and evaluate usage profiles according to policies.
3. **Neural Networks:** In KIDS, neural networks are utilized to detect attacks and locate them at a later date in an audit system. But, since it is very difficult to determine what may constitute an association or cause it, neural networks are not an efficient tool for explaining any attacks. However, they are useful for monitoring the behaviour of users within various systems. User behaviour can be predicted through experiments as discussed in [40] and using UNIX as an example, behaviours of root users are very easy to predict. When users perform automatic system actions, regular activity is monitored, making it easier to identify deviations. Few users demonstrate unpredictable behaviour.
4. **User Intention Identification:** Debar in [35] explained user intention identification as a set of approaches that allow for normal behaviours to be classified based on certain specific high-level tasks. As a result, observing system audits can help develop a list of actions. While analysing the system audit, alarms will be triggered if unusual tasks or behaviours are noted.

2.5 Machine Learning in IDSs

2.5.1 The Adoption and Motivation of Machine-Learning in Line with Attack Detection

In the modern-day world, Artificial Intelligence (AI) is recognised as centred on the key technology in a number of more innovative applications, including the identification of endeavoured credit card fraud across the area of finance, the use of a robot with the ability to detect and react to emotions, of even providing software systems with the most suitable advice

that may function as a human professional. In actual fact, there is the view that, in the event that there is no knowledge gathered from the completion of AI studies, such technologies would not exist. As discussed by[41], Machine-Learning (ML), which is viewed as a fundamental aspect of AI, is referred to as an algorithmic mechanism with the capacity to enable computer systems to learn from analogy, examples and experience. Accordingly, outputs following learning processes could be directed as intelligence in order to overcome a particular issue. Moreover, the valuable data or knowledge could be obtained from a high-volume dataset through the adoption of data-mining; in this regard, the study carried out by [42] delivers an explanation as to the definition of data-mining, highlight that it is centred on identifying anomalies, associations, changes, patterns, and events and structures of statistical significance [43], [44]. In intrusion detection, all activities and their data need to be examined in order to highlight trends in behaviours, whether normal or intrusive. Nonetheless, the point is argued that there is a need for the sample data of activities, referred to as the training set, to encompass a good number of samples pertaining to the environment under investigation so as to be able to highlight the pattern as a whole. Accordingly, new data instance could only be categorised through the learned framework in line with its similarity to normal behaviour (anomaly identification) or known attack signatures (misuse identification) [45].

A varying approach with the ability to improve the identification ability is that of ML, which is also able to achieve cost and time savings. Accordingly, as opposed to creating attack signatures or otherwise, manually outlining the more normalised behaviours of a sensor node, ML is able to perform on an automatic basis through the adoption of the most appropriate methodology and the use of a classifier. Accordingly, the requirement for human labour would not be so pronounced, and time savings could be achieved. As can be established following on from the literature review, academics promote the implementation of machine-learning alongside the mining of data so as to improve the overall performance achieved by IDSs. In the work of [46], reference was made to the significance of the role adopted by ML in terms of improving the overall capacity of IDSs in terms of placing emphasis on malicious activities through the identification and extraction of normal activities from the alarm data. In addition, it has been suggested by researchers in[46], that irrelevant alert instances are seen to decrease by as much as 99.9% following the adoption of the ML classifier.

2.5.2 Insight Identification and Standards for Data-Mining

The most suitable dataset is recognised as being one of the key considerations that need taking into account in regards to attack detection systems. More specifically, a number of different public datasets are applied in order to act as the IDS in the case of machine-learning. As has been established through a review of the literature in [47]and[48], there are two key datasets, both of which are commonly implemented in the case of network intrusion detection system; these are DARPA (Defence Advanced Research Projects Agency) and KDD Cup '99 (Knowledge Discovery and Data).

In regards the former, this is recognised as the preliminary standard corpora in the computer network attack detection systems assessment, and is gathered and accordingly distributed by MIT (Massachusetts Institute of Technology) Lincoln Laboratory in line with sponsorship from DARPA and Air Force Research Laboratory (AFRL). This particular dataset has been commonly implemented by the researcher owing to the fact that it is commonly adopted for training and testing attack identifiers where suitable modern-day results are achieved. In actual fact, a number of different datasets that form the DARPA Intrusion Detection Evaluation have been documented by the MIT Lincoln Laboratory, with 1998, 1999 and 2000 datasets utilised, with the first of these gathered for 9 weeks, notably 7 of training data and 2 weeks of testing data; the 1999 data comprised 3 weeks' training data and 2 weeks' testing data; the last of these comprise datasets across two scenarios. As has been shown throughout past works, it is not common for the DARPA dataset to be utilised following the introduction of the KDD Cup '99 dataset owing to the fact that the latter has overcome the various restrictions and drawbacks of the former. The most fundamental drawback of the DARPA is that establishing the overall accuracy of the background traffic incorporated within the assessment is not possible owing to the fact that the testbed traffic generation software is not available in the public domain. A number of other commonly cited critiques centre on the approaches applied in creating the dataset, as well as in the completion of assessments [49]. In those cases where the generation of background traffic was completed with the application of non-complex models and in the case that life traffic was utilised, there would be a notably higher false-positive rate. Furthermore, the background data did not include any factors contributing to background noise, such as strange packets and packet storms, for example. Other critiques are regards the irregularities in the data as it commented in [48].where an appreciable detection rate is shown by the trivial detector as the attacks TTL value is obviously different as well as the normal

packets. However, with all the criticisms, the DARPA dataset is slightly used by the researcher for IDS evaluation as highlighted in [50], [51].

More specifically, since 1999, the point has been made that the commonly utilised dataset centred on identification methods assessment is that of KDD Cup ’99 Dataset. This has been devised in line with the data gathered through DARPA 1998 TCP/IP. In consideration to the subset of KDD, a total of five million records are encompassed within the training data, whereas the test set is seen to comprise approximately four million records spread across a total of 41 different aspects; on the other hand, only 24 types of attack are included in each of the training data records, whilst only 14 types are added to the test data. All of the training data records are assigned with a label, either detailing the attack type or that it is normal. Importantly, the attacks are recognised as belonging to one of four different groups, including DoS, Probe, R2L, and U2R. There are detailed explanations defining the various attack types used for training purposes and these are specifically listed in[52]. Moreover, the different aspects of the dataset of the KDD Cup ’99 are categorised into one of three, as follows [51]:

1. **Basic Features:** Each of the characteristics that are able to be derived from a TCP/IP connection is contained within it. Because of this, there is a delay in the detection of attacks.
2. **Traffic Features**: These characteristics come from computations regarding window interval considerations, and there are two basic segments: similar host features and similar service features. Those connections that had the same host destination and occurred within the last two seconds are considered by the host, and those connections that have similar service and occurred within the previous two seconds are compared to one another.
3. **Content Features:** These are the characteristics that are utilized to find suspect behaviour in data. This means that features can be used to determine R2L and U2R attacks because these types of attacks are embedded in the different data portions in the packets. These typically involve one connection at a time, which is different from the DoS and Probe attacks, which examine multiple connections to different hosts in the same time period.

Nonetheless, in mind of both the cost-inducing, erroneous approach to manually classifying connections, combined with privacy-related factors, the point is made that securing public

datasets in relation to attack identification across a network is notably problematic. As such, the data of KDD has been extensively examined and quoted by the attack identification community owing to it being one of the public datasets very limitedly available.

2.5.3 Identification of KDD Cup'99 Dataset Sub-minor Attacks

There is a need to recognise that the majority of machine learning algorithms provided a suitable degree of classification for DoS and Probe attack groups owing to the fact they present a number of different connections across a short duration, whilst also providing inadequate performance across the U2R and R2L groups, with such attacks recognised as embedded in their data packets and therefore not forming a sequential pattern. This is in contrast to the cases of Probe and DoS attacks. Importantly, this means identification through any classifier is problematic[53].

When examining the various exploit groups of security, two different categories has been identified [54]:

- Remote exploit: this involves the attacking entity having the capacity to remotely connect to a machine through abusing the security threats of the network and accordingly exploiting its bugs;
- Local attack: this involves the entity being positioned to take advantage of network vulnerabilities through having an account on the local machine.

2.5.3.1 R2L Attack

Importantly, the R2L attacks are recognised as being amongst the most problematic when it comes to identification owing to the fact that they involve host-level and network-level features. Accordingly, both the network-level features and host-level features, notably the 'duration of connection' and 'service requested', and the 'number of failed login attempts', respectively, are chosen when it comes to identifying such R2L attacks[55]. When examining R2L attacks, the following attacks have been defined as R2L sub-attacks in the KDD cup'99 dataset.

- **FTP Attack**

Originally, Bhushan is the first presenter of FTP(File Transfer Protocol), which was then published in the RFC 114 [56]so as to facilitate users' transfer of files between the hosts of a network. Importantly, FTP identifies two different connections, i.e. Control Connection and

Data Connection, between the Server and Client, with the Control connection determined through the application of the widely recognised Port 21, with the data connection making use of Port 20. Notably, whilst one of the connections is concerned with data transfer, the other uses such data. When the FTP protocol is attacked through the attacker utilising the PORT command with the objective to garner indirect access to ports, this is referred to an FTP bounce attack. Notably, this makes use of the machine belonging to the victim, which is essentially positioned as a middle-man, handling any requests. Such an approach is directed towards the discrete scanning of port hosts, as well as when seeking to garner access particular ports otherwise unable to be accessed by the attacker via a direct connection; the Nmap port scanner provides a valuable example in this regard. Importantly, the vast majority of current FTP server programs are configured in such a way so as to routinely reject PORT commands that could potentially create links to any host besides the original one; this, therefore, achieves success when it comes to preventing FTP bounce attacks[56].

- **Password Guessing**

One of the most widely successful forms of attack is through password-guessing, which involves attackers successfully guessing a password from a local or remote position, with the use of either an automated method or otherwise manual approach. This means of gaining access is not nearly as difficult as might be first considered. The majority of networks are not configured so as to require complicated passwords; as such, network access can potentially be achieved through an attacker identifying just one weak password. Moreover, when it comes to guessing attacks, not all authentication protocols are correspondingly effective: as an example, owing to the fact that case insensitivity is inherent in LAN Manager authentication, password-guessing attacks do not need to take into account whether or not passwords letters are lowercase or uppercase, for example.

- **IMAP Attack**

IMAP (Internet Message Access Protocol) is recognised as an everyday email protocol providing email message storing facilities on a mail server, whilst also delivering the end-user with the ability to view and make changes to messages in much the same way as if they were stored locally, i.e. on the user's own device. The majority of IMAP implementations provide the facility of multiple logins, which importantly positions the end-user in being able to connect to the email server through different devices at one time. Although IMAP is known to offer an authentication facility, nonetheless, there is the potential for it to

be relatively simple overridden by any individual with the ability to steal passwords through the application of a protocol analyser; this is possible owing to the fact that the username and password of the client are communicated as clear text. In the case of an Exchange Server setting, it is possible for this security flaw to be managed by administrators, notably via SSL (Secure Sockets Layer) encryption for IMAP. [57]

- **Warezmaster Attack**

The WM (Warezmaster attack) is able to take advantage of FTP server misconfigurations, with the majority of FTP servers known to support the ban-anonymous FTP approach, which enables users to garner access to files without ever be requested to identify themselves. When the server facilitates anonymous login, users are then well-positioned to log in using the username 'anonymous'; it is commonplace for the password to be provided by the server. When this has been done successfully, the Anonymous FTP can then be directed towards downloading or otherwise gaining access to those files that are publicly available from the server. The FTP server is commonly configured in a way that anonymous users such as these cannot make use of write permissions. If, however, written permission is granted to a user through erroneous configuration, the attacker is then able to log on to the server using the anonymous credentials, meaning hidden directories can be created and large files uploaded to the service. This is what defines a WM attack. [58]

- **Warezclient Attack**

The WM attack has been further developed to become the WC (Warezclient)attack, which is recognised through any anonymous/legal user being able to download the malicious files uploaded onto the server by the attacker. Such an attack may induce a number of different outcomes in regards to the host machine, where outcomes notably rely on the type of Warez uploaded.[58]

- **Spy Attack**

A spy attack is recognised as the practice of gaining access to information without the consent or knowledge of the information-holder, with access gained through competitors, enemies, governments, groups, and individuals for economic, military, personal or political advantage, notably through the use of individual computer, internet or network methods, utilising cracking approaches, malicious software and/or proxy servers. This form of attack through spying commonly involves access to classified information, the control of whole networks or

individual computer systems in order to achieve a competitive edge and for sabotage, and physical, political and psychological subversion activities.

- **PHF Attack**

Common Gateway Interface (CGI) is recognised as an approach with the capacity to call external software through a web server with the objective of providing dynamic content. In the specific instance of a CGI program, 'phf' seeks to remove dangerous characters and accordingly pass such strings to shell-based library calls. As an example, URL likes '/cgi-bin/phf?Qalias=%0a/bin/cat%20/etc/passwd', for example, could achieve the user/password content of /etc/passwd on the target host.

- **Multi-Hop Attack**

When there is a hack into a mail server which subsequently garners access to a client where the email server is on the same server, this is recognised as a multiple-hop attack path. Despite the fact that the internet may not, at that time, be being used by the client, nonetheless, an attacker can follow a specific attack chain or path in order to gain access to the vulnerable target.

2.5.3.2 U2R Attack

Semantic information, which is commonly problematic to capture throughout the more preliminary stages, is required in the case of U2R attacks. This type of attack is commonly content-based, with an application targeted[55]. Accordingly, when there is a U2R attack, various aspects, including the number of shells prompts invoked or the number of file creations, are chosen, whereas there is the disregard of other features, including source bytes and protocol, for example. In KDD Cup'99 dataset the following attacks are defined as U2R sub-attacks;

- **Buffer Overflow Attack**

The Buffer Overflow attack is witnessed when an effort to write more data to a specific memory block is attempted by a process or program, which notably relies on the dst_byes feature, which relates to the amount of data bytes identifiable between the destination and source. When the latter is seen to be higher than normal, a Buffer Overflow attack is witnessed.

- **loadmodule Attack**

A loadmodule attack is seen when there is the case of a User to Root attack against SunOS 4.1 systems, which are known to utilise the windows system referred to as xnews [29]. Owing to the presence of a bug in the way in which the loadmodule program completes the sanitisation

of its environment, it is possible for the local machine to be accessed at the root level by unauthorised users. In this vein, it is stated by[59], should there be dst_bytes equal to between 186 and 1696, a loadmodule attack may then occur.

- **Perl Attack**

In the case of a Perl attack, this may arise when there is the setting of the user ID to root in a Perl script, with a root shell phfR2L Exploitable CGI script created, which subsequently facilitates the execution of arbitrary commands by a client.Notably, when such an attack is identified, the 'root' feature is seen to have a value of 1. [60]

- **Rootkit Attack**

A rootkit is explained as a hidden computer program with the aim of delivering ongoing privileged access to a computer whilst ensuring its presence remains hidden. The concept of 'rootkit', in this regard, is seen to refer to a link between the words 'root' and 'kit'. Formerly, a rootkit was recognised as a number of tools facilitating access to a network or computer on an administrator level basis. In this case, the root relates to the Unix and Linux system Admin accounts, whereas the kit is seen to relate to the software components making use of the tool. At the present time, rootkits are more commonly seen to be linked to malware, which is known to mask their presence and actions from system processes and users. Importantly, a rootkit enables someone to main control over a particular system without its presence being identified by the user[61]. Upon the installation of a rootkit, the rootkit controller is able to change the configuration of the system on the host machine, and can further execute files. When a system is infected, a rootkit is also able to spy on the use demonstrated by the genuine user, whilst also gaining access to log files.

2.5.4 Machine Learning Detection Approaches

2.5.4.1 Data Labels

In consideration of the dataset label availability, two different functioning modes are recognised as identification approaches, as highlighted by [51]. First, supervised the detection, through which the predicted framework is created in line with the labelled training set comprising samples that are both anomalous and normal in nature. The point is made that, through this approach, the rate of identification will increase as a result of information access capacity. The second approach is that of unsupervised detection, where no training data is

necessary when there are unsupervised techniques owing to the fact that this method is centred on two underlying expectations: first, that normal traffic is characterised by the majority network connections, whereas malicious traffic is characterised by the minority of connections; and second, that, from a statistical standpoint, there is a difference between malicious and normal traffic. Therefore, the instances that appear infrequently and are significantly different from most the instances are considered as attacks while the normal traffic is reflected by the data instances that build groups of similar instances and appear very frequently are supposed.

2.5.4.2 Output Format

In line with the way in which anomalies are reported, more commonly, the output is seen to fall into one of three types, as highlighted by [51]namely scores, binary labels, and multi-labels. The scores method takes each tested instance and assigns a numerical score in order to establish the likeliness of the attack, meaning that a ranking approach is applied in order to categorise the most significant samples by outlining the threshold value, with Naïve Bayes providing a good example of such an approach. In the case of the binary label, some of the identification approaches are unable to detail instance scores, but instead opt to apply labelling, where the instances undergoing testing are either labelled anomalous or normal. Secondly, the multi-label approach assigned each instance undergoing testing with a particular label, with one label for normal traffic, whilst attacks receive their own corresponding label, such as DoS or Probe, for example, with such an approach utilised when the instance cannot be scored, such as in the case of the Decision Tree approach.

2.5.4.3 Classification Techniques

This thesis directs its emphasis onto multi-class classification issues, as will be highlighted later on in this work. Classification algorithms when applied in the attack identification are applied in order to complete the categorisation of network traffic as either normal or an attack. In essence, following the presentation of anomaly identification by Denning [62]a number of other approaches have been suggested. Accordingly, a number of the more commonly implemented techniques in dealing with the multi-class issue is mentioned by the studies of [51], [63]–[66] as follows;

- Bayesian Networks are recognised as presenting a probabilistic graphical framework through which the concept is centred on the illustration of a number of different factors and their corresponding probabilistic independencies. This particular method, through acyclic graphs, comprises both edges and nodes: edges are seen to program conditional

dependencies between factors, whilst nodes represent the factors. These are adopted in categorisation and identification in a number of different ways. Importantly, the most commonly implemented methods of this type are Bayes Network and Naïve Bayes.

- Clustering: Such an idea is based on the concept of unsupervised identification, meaning, should the two assumptions made, as detailed above, be found to be true, anomalies can then be identified in line with the cluster size, i.e. large clusters are aligned with normal data, whilst the remainder are aligned with attacks.
- Neural Networks comprise a number of different computational units, where there is the adoption of a complicated mapping operation between such units. Primarily, the network is trained through the use of label datasets, meaning the instances under testing subsequently undergo categorised as either attacks or normal following network fading. SVMs (Support Vector Machines) and MLP (Multi-Layer Perception) present Neural Network approach examples and are commonly adopted in the case of anomaly identification.
- Trees adopt a flowchart-type tree structure, which is created when nodes represent the features, whilst testing can be seen to be represented by the branch whilst leaves signify predicted classes. A number of different approaches fall within this group, with the most commonly implemented for the categorisation issue including Random Forest and J48, for example.

2.6 Summary and Conclusion

This chapter has presented an overview pertaining to the security of Cloud Computing systems, coupled with IDSs, in addition to attack categorisation. It is noted that with regards to the implementation of IDSs, the user in Cloud Computing is recognised as a requirement and accordingly discussed in this regard. In relation to IDSs and machine-learning, there has been a review provided in regards the datasets publicly available, with an examination into the widely used KDD Cup ’99 dataset.

As can be seen upon reviewing the literature, a number of different academics and scholars in the field have proposed the use of data mining and machine learning in view of achieving IDSs performance improvement. Machine-learning is recognised as encompassing various identification approaches. Accordingly, in an effort to decrease the study scope, supervised identification will be taken into account with the adoption of the KDD Cup ’99 dataset owing to the fact that, as in the case of supervised dataset, it is possible to complete an analysis on

each of the attack labels in consideration to its various features and behaviours. In line with the way in which anomalies are reported, this thesis will implement labelling output, with consideration directed towards both multi-class and binary classification owing to the fact that there is a need to take into account various attacks and factors, which also warrants the application of different categorisation methods in order to improve identification and to further examine the factors underpinning low levels of accuracy in identification. This is done owing to the fact that all categorisations have the potential to improve or identify particular attacks in line with the parameters available and also in line with attack trends and mechanisms.

Chapter 3 : Developing an Understanding of the Classification of Imbalanced Datasets

3.1 Introduction

This chapter investigates the various challenges and considerations witnessed with regards to the identification and classification of network attacks. Furthermore, the classification approaches will be investigated in the presence of an imbalanced dataset, with the various methods centred on learning from the imbalanced data typically used in practice.

3.2 KDD Cup '99 Dataset Classification Challenges

When utilising KDD Cup '99 dataset, as has been previously detailed, class imbalance and duplicate records are amongst the most prominent challenges. There are a number of different duplicate instances in this particular set, notably as a result of the lack of temporal data. Accordingly, data quality is impacted, with the machine-learning methods' and training processes negatively affected. Researchers in [67], [68] conclude that there is a need for diversity amongst the training samples. In a similar vein, the effects of duplicates on the utilisation of Naïve Bayes and a Perceptron with margins has been the focus of an empirical investigation carried out by [68], adopted in the identification of spam, with the researchers suggesting the complete removal of duplicate instances, recognising the volume of duplication as having a notably negative impact on classifier accuracy. As can be seen in Table 3-1, each class's number of instances both prior to and following duplicate removal has been detailed. It highlights that most duplicates as being present in the case of Probing and DoS classes, predominantly owing to the nature of intrusions. Furthermore, it is apparent that this particular dataset experiences problems as a result of the class imbalance, as shown in the Table3.1 when recognising the fact that the Prob and DoS attacks enhance the most number of samples and also mentioned in the study of [68], [69].

Table 3-1: Description of class distribution in KDD Cup 99 dataset

	DoS	R2L	U2R	Probe	Normal	Total
KDD Cup 99 dataset (with duplicates)	3883370	1126	52	41102	972780	4898430
KDD Cup 99 dataset (no duplicates)	247267	999	52	13860	812814	1074992

3.2.1 The Challenges Caused by Imbalanced Datasets

Imbalanced datasets are, in the classification problem domain, identified on a regular basis, with this particular issue stemming from the significant imbalance in a number of examples between one class and another. Which mean a greater misclassification rate will occur in the

case of the minority class owing to the reason that standard classification learning algorithms commonly demonstrate bias in relation to the majority class. Accordingly, there is a need to direct attention to this particular issue owing to the position of many that minority class is commonly representative of the most fundamental concept for learning[70]. Nonetheless, owing to the fact that minor classes' data gathering is expensive or is otherwise linked to notable cases, as highlighted in the work of [71], it would appear that represented samples are problematic to acquire. In most cases, binary classification is seen to pertain to the imbalanced classification issue; however, other issues arise in the case of imbalanced data called multi-class classification. In this case, the point is laboured in the work of [4] that, when it comes to overcoming this issue, it proves to be more problematic owing to the need to balance a number of different minor classes.

In the case of imbalanced datasets, the problem is that a number of different machine-learning algorithms, such as DT and ANN, for example, demonstrate bias in relation to the major classes—which is a point highlighted in the work of [71]. Accordingly, this can result in the poor classification of the minor classes, with the minor class commonly ignored by a number of different classifiers as mentioned in the work of [72], [73], although a significant overall accuracy is still achieved [4], [69]

During more recent times, the issue of imbalanced learning has been the focus of much attention and focus in terms of research effort. In actual fact, a number of different application domains have sought to manage the issue of class imbalance as a fundamental consideration warranting attention. Some examples include, predicting ozone levels, as demonstrated in the work of [63], face recognition[74], and also in the critical field of medical diagnosis[75], [76], inter alia medicine [77], [78], chemistry and biology [79], the processing of natural language [80], lexical acquisition [81], text recognition [82], and attacks and fraud identification [83].

3.3 Managing the Challenge of Attack Classification in the Presence of Imbalanced Dataset

A number of different techniques have sought to present suggestions in order to manage imbalanced classification, with such recommendations able to be grouped into two different categorisations, namely algorithm level, and data level[68][69]. In the case of the former, such methods are also referred to as internal approaches owing to the creation of a new algorithm, or a present one amended so as to manage the problem of imbalanced datasets as highlighted in [84]–[87]. In regards to the data level, this is also commonly referred to as external

approaches, where data undergoes pre-processing in order to eradicate class imbalance problems. Within this group, the training data is provided with some sampling form. However, notably, a number of class imbalance methods are implemented both at the algorithm level (internally) and at the data level (externally) methods. As aiming to achieve minimisation in regards minor class misclassification samples' costs as presented in the work of [88]–[91]

A number of other methods are recognised as based on ensemble learning algorithms, which are implemented through the incorporation of a cost-sensitive model(algorithm level) within the learning process [91] or otherwise via data pre-processing prior to the initiation of the learning phase for all of the classifiers as presented in [92]–[94].In this regard, throughout this part of the study, the sampling methods are presented. Following, the cost-sensitive learning technique is then discussed. Lastly, a number of valuable and pertinent ensemble methods within the model of imbalanced datasets are discussed.

3.3.1 Data Level: Resampling Techniques

The application of a particular type of sampling is recognised as one of the most widely implemented approaches to managing class imbalance in the case of training data. In regards more specialised literature as in the work of[95]–[97], it has been confirmed that the implementation of a pre-processing stage with the aim of achieving class distribution balance is most commonly recognised as a valuable solution. Such resamples methods are broken down as belonging into one of three groups: the first seeks to take away instances from the major classes with the aim of securing the outcome of a balanced dataset, which is referred to as under-sampling. The second seeks to create a number of additional minor class instances so as to achieve dataset balancing and is referred to as over-sampling; and the third is a combination of the aforementioned two methods, which utilises both under- and over-sampling, which is recommended for imbalanced datasets in the study by [71].

In actual fact, it is possible for resampling to be carried out randomly, based on used approaches and the used dataset. At times, there is the suggestion of random sampling owing to it being able to create a satisfactory outcome. However, the writers in[4], [69]have presented the view that random sampling could potentially induce issues owing to the lack of consideration towards data distribution. When implementing random under-sampling, the point is made that valuable data could possibly be removed, which would then have an impact on the learning process. Furthermore, when it comes to the biased problem undergoing random over-sampling, this could potentially mean a greater increase in bias owing to the fact that an exact replication

of the existing instance is made, meaning there will be an increase in over-fitting. Nonetheless, it is recognised that both sampling methods are valuable and have their uses as in the study carried out by [98], [99]. Nonetheless, class distribution in the case of resampling has been taking into account through various approaches, such as synthetic training samples generation, as in the work of [100], with 'Synthetic Minority Oversampling Technique' (SMOTE) noted as one of the most sophisticated approaches, which is based on the concept of incorporating a number of different instances of minority class that are seen to reside together so as to generate new minor classes for the purpose of training set over-sampling. Nonetheless, the actual distribution of a real issue is not always possible to determine as presented in the work of [71], with a number of other scholars in the field like the work in [73], [101]recognising that this is centred on the approach and the issue.

Researchers in [102]presented a sampling method centred on clustering, which sought to take the disjunct of minor class and implement over-sampling, requiring the organisation of the data with significant characteristics into groups from the training data, with them over-sampling and/or under-sampling then implemented. In the study carried out by [103], the cluster with a C4.5 DT and an MLP trained with backpropagation is implemented, with the outcome providing support for cluster adoption. A number of other valuable examples include Cluster-Based Oversampling (CBO as discussed in [98], as well as that of Class Purity Maximization [104], Sampling-Based Clustering [105] and the agglomerative Hierarchical Clustering [77], amongst others. Nonetheless, the majority of the works taken into account and discussed examine the most appropriate resampling approaches for adoption with the aim of enhancing classification algorithm behaviour in the case of imbalanced datasets, and accordingly restricting investigation for, in the main, classification issues as discussed in the work of [69], [106].

3.3.2 Algorithms Level: Cost-Sensitive Learning

During more recent times, academics and professionals in the field have examined the factors underpinning inadequate learning amongst a number of different machine-learning approaches from imbalanced datasets; with a number making the point that assessment criterion is one of the key considerations. Since the overall accuracy is the most used metrics to evaluate the performance of the classifier in spite that the overall accuracy, mostly, not present the biased of classifier towards the major class. In which the minor class(es), sometimes, is ignored by the classifier especially in the case that the extreme imbalance dataset is used. A number of researchers have examined bias amongst classifiers, drawing the conclusion that DTs and

ANNs provide key examples of such classifiers [107]. A number of different alternative evaluation metrics, such as AUC1, F-measure and weighted metrics (cost-sensitive learning), for example, are all taken into account and reviewed in the works of [71], [72] with the most commonly implemented method seen to be the cost-sensitive learning approach, which involves the multiplication of the classification or error rate for each objective by a cost/weight. Otherwise stated, this approach is centred on the concept of a weight matrix, which is seen to be aligned with the confusion matrix garnered following classifier performance, where the most appropriate classification does not induce any form of penalty. There is the suggestion that domain knowledge should be incorporated into such a weight matrix; this is applied in the case of the KDD Cup '99 dataset, as an example, as can be seen in Table 3-2. Owing to the fact that its application is centred on assessing the classification result, the class imbalance issue is not addressed. In this regard, the focus is instead directed towards penalising the misclassifications of U2R and R2L instances, which is suggested by the study in [4], [108]–[110] During more recent times, a number of approaches have been implemented in regards cost-sensitive learning, as shown in the work of [111], suggests its adoption for weighted rough sets.

Table 3-2: Weight matrix for evaluating the result of the KDD cup 99 competition[109]

	Normal	**Probing**	**DoS**	**U2R**	**R2L**
Normal	0	1	2	2	2
Probing	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

3.3.3 Classifier Combination: Ensemble Learning

Ensemble-based classifiers, which are recognised as being built on the concept of creating at least two classifiers from original data, aggregate predictions when there is the presentation of unknown instances, with such classifiers created through bringing together a number of different classifiers in order to generate a new classifier with the potential to exceed in terms of classification ability. More specifically, data or algorithm levels and ensemble learning methods are utilised in combination. In regards the data level, data pre-processing is carried out, with each classifier then undergoing training; in regards cost-sensitive ensembles, on the other hand, the ensemble learning algorithm is applied in such a way so as to direct the cost-minimisation process [112], [113]

A. Bagging

Bagging is recognised as one of the first ensemble algorithms, and is seen to be simple and lacking any degree of complexity so as to ensure efficiency. Bagging, which is the abbreviation given for Breiman's bootstrap aggregating method, is centred on the application of training data's bootstrapped copies in the creation of a number of different results, where large data subsets as acquired from the training data are drawn on a random basis with replacement. Such subsets are then directed towards creating a framework comprising individual classifiers. In line with majority voting between specific classifiers, there is then the initiation of the ensemble decision. One method deriving from bagging centred on the adoption of various decision trees in order to build a model, with this particular method referred to as Random Forest; this, unlike bagging, utilises feature subsets that are focused on the random subspace method.

B. Stacking

Wolpert's stacking generalisation is concerned with achieving classifier performance improvements, with the key factor underpinning misclassification recognised as the class decision boundary being closely located to the neighbour class. Furthermore, the classifier used positions it on the incorrect side of the boundary. When implementing stacking, classifiers' ensemble output is utilised as the second-level meta classifier input (Wolpert, 1992). In the case of a number of different studies carried out in this regard[94], [114], [115] and[116] the ensembles of classifiers has been highlighted as one of the approaches suggested in order to overcome the issue of class imbalance. In the case of the work by [97], however, the adoption of training single classifier and data processing is recognised as achieving sound results, whereas other researchers emphasise that the most simple approach to attaining sound performance are through RUSBoost[117] or UnderBagging [69], [118]

3.4 Feature Selection on Imbalanced Dataset

3.4.1 The Need of Feature Selection on Imbalanced Dataset

When completing high-dimensional data analysis, the process is recognised as challenging and oftentimes problematic for workers in the field of data-mining and machine-learning. One valuable and effective approach to solving this issue is through feature selection, which is achieved through the removal of redundant and irrelevant data; this facilitates time decreases in computation, enhancing learning accuracy, and further encourages greater insight into the data or learning model.

In defining the concept of feature selection, it is stated that the process centres on securing a subset, as derived through an original feature set, with consideration to criterion outlined in regards feature selection. This helps to facilitate the choosing of the most valuable aspects of the dataset. The issue of feature selection has been recognised and widely acknowledged for more than four decades and is well known to play a key part in data processing scale compression, with the more irrelevant and redundant of features removed. The overall approach of feature selection has the capacity to pre-process learning algorithms, where sound feature selection outcomes can help to enhance overall learning accuracy whilst making learning results more simplistic and learning times shorter [119], [120]. One notable aspect is a feature space is seen to be of significance and value to the class when it provides valuable insight into the class and whereupon classification performance is degraded through its removal. The irrelevant feature is that which does not present any valuable data relating to the class and where its presence is seen to elicit classification performance reduction [121]. One of the most irrelevant aspects is a noisy or redundant feature, for example, where the latter would be unable to deliver valuable information in line with classification following the choosing of the most optimal and appropriate subset of features; this is owing to another feature already providing this same data. In the case of a noisy feature that is not considered redundant, information relating to class is not present.

Importantly, the studies in[122]–[124] conclude the two ways via which dimensionality reduction can be achieved are through feature selection and feature extraction. In contrast to feature selection, in the case of feature extraction, there is commonly a need to implement original data transformation to features with a key pattern recognition capacity, where the original data is considered to be featured with a weak recognition aptitude. When considering feature selection-which has long been recognised as a study topic in methodology-its use has been witnessed in a number of different fields, including in the fields of image-recognition[125]–[129],image-retrieval[130],[131],intrusion-detection[132]–[134],text mining[135]–[137],bioinformatics-data-analysis[138]–[143],fault-diagnosis[144]–[146]and so on. According to the theoretical-principle, feature selection methods can be based on statistic[147]–[150]information-theory[151]–[156] manifold [157]–[159] and rough set [160]–[164], with all of these is able to be grouped in line with different standards.

1. In line with the training data assigned, i.e. labelled, unlabelled or partially labelled, the method of feature selection is broken down into supervised, unsupervised and semi-supervised.
2. In line with the link between feature selection method and learning method, the former is categorised into embedded, filter and wrapper models.
3. In line with the evaluation criterion, the methods of feature selection may result from consistency, correlation, dependence, Euclidean distance, and information measure.
4. In line with the search methods, the approaches to feature selection could be broken down into backward deletion, forward increase, hybrid, and random models.
5. In line with the output type, the approach to feature selection is broken down into subset selection and feature rank/weighting models.

3.4.2 Approaches to Feature Selection

3.4.2.1 The Filter Approach

Feature selection's filter approach decreases the total sum of features utilising the data's properties to the learning algorithm actually implemented [121]. One key benefit identified in the adoption of a filter algorithm alongside a feature set can be seen when considering the reduced number of features implemented in the ultimate induction algorithm. Accordingly, classification algorithms will demonstrate improvement, alongside a reduction in computer processing time. In contrast to the wrapper approaches, filter methods, in their process, are not inclusive of the ultimate learning algorithm. Such independency has been recognised as a further advantage associated with the adoption of filter methods as presented in [165]. One further identified advantage is that the same aspects could be applied in other learning algorithms for the purpose of comparative analysis. In this vein, it is noted in the study by [166]that a number of filter algorithms, including, for example, Correlation-based Feature Selection (CFS), could present findings comparable to or an improvement on wrapper models in various aspects. In this vein, a new correlation-focused selection approach was presented in the work of [167], with the research highlighting the overall effectiveness and efficiency of these approaches in the management of highly dimensional sets of data. Nonetheless, as has been recognised by [168], filter-based selection techniques present the drawback of failing to interact with the classifier algorithm ultimately applied. One further drawback emphasised is that the majority of the filter approaches have a notably univariate nature; this is taken to infer that they do not take into account other aspects and the qualities of such. Importantly, the work was completed on a highly dimensional bioinformatics data set as mentioned by [168].

The filter-based feature selection approach was benchmarked against 15 different sets of data, with this also done for one wrapper approach, through the completion of experiments by [169]. The conclusion drawn by the scholars stated that filter-based approaches differed from one to the next, with such differentiation stemming from the data set itself; overall, however, they were seen to be quicker, whilst also garnering improvements in terms of algorithm classification effectiveness.

Three different filter algorithms, notably two multivariate algorithms, namely Correlation-based Feature Selection (CFS) and Relief-F, and then information gain, which is a univariate algorithm, will be assessed here. The underpinning of the Relief-F algorithm, as presented by [170], is the ability to choose features on a random basis, and accordingly—in consideration to the closest neighbours—assign a greater degree of value to those features distinguishing between classes. Such aspects are subsequently graded in line with their relevance. In the study by[171], which was empirical in nature, the conclusion was drawn that comparable findings were garnered through the application of the Relief-F algorithm when compared with those of other filter algorithms, namely Gain Ratio and Information Gain, for example, when the Relief-F algorithm is applied in their specific field.

Correlation-based Feature Selection (CFS) algorithms seek out features that achieve significant levels of correlation with the class, which are seen to have no correlation with one another—or only a very minor correlation[172]. The most recent feature selection algorithm is that of information gain (IG), which is defined as an approach concerned with weighting features in line with a relevancy score, where such a score depends on all respective attributes. The correlation between attributes is neglected, which therefore positions the approach as univariate. There has been the completion of comparable works, considering Gain Ratio and CFS approaches, in line with various data domains. It was established in the work of[173] that, when applied, the CFS approach provides more valuable outputs than the Gain Ratio, although this induces significant costs in terms of computer time.

3.4.2.2 The Wrapper Approach

In contrast to the filter approach, wrapper algorithms take a preselected induction algorithm as one aspect of the feature selection approach. Along with the additions or exclusions of features, the ultimate results are then graded in terms of selection effectiveness. Owing to the fact that the induction algorithm is used throughout the assessment stage of the selection process, the wrapper approaches are better positioned to achieve improved results than the filter methods.

In this vein, the wrappers for feature subset selection were contrasted alongside filter approaches as in the work of[174],with the conclusion drawn that attribute relevance are significant contributions in line with the learning algorithms' performance when the algorithm is viewed and accounted for. Nonetheless, there are a number of different restrictions to such approaches. The computational cost of running such an assessment is recognised as far more significant than that identified when applying a filter method, with such costs increasing in line with an increment of attributes. One further drawback of the wrapper approach can be seen when considering the potential of data over-fitting.

Notably, there are also other types of wrapper approach. As opposed to applying an individual method wrapper, as in the case of sequential forward selection, for example, a new approach is presented in the work of [175], namely simulated annealing generic algorithm (SAGA), which is seen to combine present wrapper approaches into a single solution. The study has emphasised that incorporating other approaches allows the drawbacks inherent to each individual method to be decreased if not altogether eradicated.

In the study of [176], a wrapper method in line with the Support Vector Machine (SVM) classification was provided. The conclusion of the work stated that the application of such an approach would ensure data over-fitting would be circumvented as a result of its ability to achieve data splitting. It further enabled the application of various Kernel functions in achieving more optimal results. One disadvantage identified showed that the suggested algorithm utilised the backward elimination aspect, which was found to be very costly from a computational perspective when utilising highly dimensional sets of data.

3.4.2.3 Hybrid/Two-Stage Design

An approach that brings together the above-discussed methods has also been suggested in the work of [177], [178], with this approach utilising a filter approach with the aim of eradicating irrelative aspects, and then a classifier-specific approach to further decrease the feature set. Through ensuring the feature set is decreased from n features to achieve a lower number k, there is a reduction in the computation space in relation to the number of features—notably from $2n$ to $2k$. Such a combined filter-wrapper approach is able to exploit the advantages associated with the use of the wrapper model whilst simultaneously reducing the computational drawbacks associated with the use of the wrapper method in isolation.

In order to manage and better handle the previously highlighted disadvantages and to further circumvent the problem of having to outline a stopping criterion, a number of different scholars

in the field have sought to make use of the benefits associated with the wrapper and filter approaches. Notably, both an independent measure G and a fitness evaluation function of the feature subset *A*, are adopted through the application of hybrid algorithms, with the knowledge provided by a filter algorithm and a particular machine-learning algorithm then utilised in such a way so as to efficiently select the most optimal subset of feature S_{best} [179]. Importantly, when applying a hybrid algorithm approach, the search is begun from an empty subset S_0, with the process then repeated to identify the most optimal subset. Across all iterations, when seeking to determine the best subset of features with cardinality k, all of the potential subsets of *k+1* are examined; this is achieved through the incremental addition of features from those that are remaining. The independent measure, *G*, is applied in such a way so as to assess each of the subsets that have been generated, S, which are seen to incorporate the cardinality *k+1* and further draws a contrast with the previous most optimal subset. Should it be the case that *S* is seen to be more superior to the previous most optimal subset, it would then be viewed as the present most optimal subset S'_{best} with k+1 cardinality. When there is the conclusion of the iteration and the identification of the ultimate S'_{best} at level *k+1*, there is the implementation of the fitness assessment function, A, to the S'_{best}, with the evaluation outcome then contrasted alongside that of the most optimal subset identified at level *k*. Importantly, when there is no further improvement, meaning the very best subset is secured by the hybrid model, the process of searching for the most optimal subset S_{bet} is then ended. As has been highlighted in[180], approaches inherent in this particular category are not as time-efficient as the filter methods; nonetheless, they are far more efficient and are able to attain improved performance in terms of classification.[181]

3.5 Related Works

3.5.1 Imbalance Learning Methods

3.5.1.1 Navies Bayes

Navies Bayes (NB), is the more simplistic version of the Bayesian Networks (BN) approach and is recognised as centred on the features independence hypothesis. Accordingly, a one-dimensional kernel density estimation is achieved following the reduction of high-dimensionality density estimation. In this case, the NB training time is initiated in linear time. In contrast to that of BN, NB is recognised as being less expensive, from a computational standpoint, due to there being no need for a priori knowledge in regards the issue when striving to establish probabilities.[4]

In the 2004 work by [182], NB and Decision Tree (DT) performances, notably adopting the KDD Cup '99 dataset, are contrasted in terms of performance, with DT achieving the greatest accuracy, whereas NB is seen to demonstrate improved performance in the case of minor attacks identification. A comparable conclusion has been achieved in the study by[183], recognising that ANN does not demonstrate as high a level of performance when examined in line with minor attacks, whereas a low error rate is scored by NB for such types of attack. In line with a number of the experiments carried out in the works of [107]and[98], it is stated that ANN and DT are biased towards major attacks, meaning that there is an overly high false-positive rate; notably, this is not the case with NB. Nonetheless, such experiments are implemented across imbalanced datasets and ensemble learning approaches.

One further work carried out by [184]presents the conclusion that, when it comes to identifying minor attacks, NB is the stronger method. In this case, the performance demonstrated between the two probabilistic methods (NB and Gaussian classifier) and two predictive techniques (DT and RF) undergoes comparison. The conclusion is drawn that the two probabilistic approaches illustrate a higher level of performance in the case of minor attacks; however, in the case of DoS attacks, identification by NB is seen to be low. A number of other experiments focused on drawing a comparison between the Adaptive Bayesian Network and NB: as a result of low minor attack samples, a lower identification rate was achieved by ABN in regards the minor classes; on the other hand, a high identification level in terms of accuracy was achieved in regards the major classes.

During more recent times, possible combinations of methods have been implemented stemming from a number of observations made in various works, including in the cases of [182], [183], as highlighted earlier. Furthermore, in the study by[185], the false-negative rates have been the point of attention, along with a hybrid framework focused on irregularity identification and misuse identification. Accordingly, irregularity identification is utilised in order to define the normal traffic. It is recognised that NB demonstrates a much better performance than DT in the case of attack identification. Other works[186] emphasise that a hybrid model should be carried out and should incorporate two different levels, where the first should focus on the implementation of Self Organising Map for normal instances, whilst the second level should present NB. A hybrid system is recognised as achieving higher classification rates than that which can be garnered through NB in isolation. Nonetheless, through the completion of such works, the dataset is pre-processed and not balanced.

3.5.1.2 Ensemble Learning

Owing to the fact that it can be difficult to arrive at the most appropriate and accurate hypothesis, ensemble algorithms are recognised as being the most valuable approach. Owing to the fact that the ensemble approach brings together various hypotheses, a more promising and valuable outcome will be achieved than if just one hypothesis was used in isolation.

Outlier identification positioned at the front end of the existing system was implemented, as in the study by [187] in the instance that abnormal traffic is identified, the data would then undergo categorisation to one of the KDD cup '99 attack groups through the application of the RF algorithm. In the case of this work, there is the generation of balanced data owing to the fact that the instance with the least occurring attacks is replicating. The conclusion is drawn that, when implementing the original data, the false rate is greater than if the balanced data is utilised. Despite the fact that this work is seen to attain sound findings through error rate minimisation, it remains that the approaches to data balancing are not concerned with resampling and similarly do not place emphasis on ensemble within classes. One further work in [188], which have adopted a similar method, is that of [187] however, the work also implemented normalisation methods throughout the pre-processing. This work has attained significant accuracies across major attacks, whereas minor attacks demonstrated low identification accuracies (5% for R2L and 35 for U2R).

A research focused on achieving enhanced classification accuracy and a lower degree of false positives was carried out in the work of [189], which sought to achieve good results in the NSL–KDD dataset classification, utilising approaches of bagging, boosting and stacking ensemble. The conclusion was subsequently drawn that, when examining the false positive rate, the greatest reduction was achieved by the stacking approach. Importantly, however, this work implemented ensemble without the balancing of the dataset.

The bagging schema was implemented in the study of [190], with a contrast drawn between six different binary classifiers, namely a bagged family of C4.5 classifiers, a bagged family of naïve Bayes classifiers, a bagged family of PART base classifiers, C4.5, naïve Bayes and partial decision tree classifiers (PART). This work has established that C4.5 without bagging demonstrates greater performance than a bagged PART ensemble. In the examination of training time and classification accuracy, bagged ensembles were not found to achieve greater performance than the individual base classifiers. The model time was reduced via the deployed approaches, without placing emphasis on the accuracy of attacks.

3.5.1.3 Feature Selection and Resampling

As can be seen, when reviewing the literature, there are a number of different methods with the potential to eradicate the issue of class imbalance, with the inclusion of internal approaches that customises algorithms to data selection approaches imbalanced data, ensemble learning, and cost-sensitive learning. Although it is possible for the interested reader to identify a summary of approaches to class imbalance learning and feature selection in imbalance domain in [70], [191], it is nonetheless recognised that some attempts[192]–[194] have been concerned with the examination of the joint influence of resampling and feature selection when it comes to attempting to manage class imbalance. In the study of [193], the findings underwent an examination with the utilisation of under-sampling approach or feature selection, taking into consideration datasets associated with the prediction of software quality. The empirical data garnered throughout the work emphasised that, when applying feature selection on the data sample, there is a greater level of performance than when choosing the features on the original data. Nonetheless, this particular study was carried out in consideration of a specific field. Furthermore, the researchers only took into consideration the testing of random undersampling and various techniques of filter feature selection (e.g., $\chi 2$, Relief, Gain Ratio). In the research by [194], the performance demonstrated by a number of different feature selection metrics was analysed, in addition to the way in which class imbalance was overcome. In addition to the use of 7 filters as a feature selection approach, the researchers further completed an analysis on the sample methods' performance. SVM was applied as base learning, whilst 10 publicly available datasets presented the sample for experiments. The findings emphasised that the correlation coefficient of Signal-to-Noise and Feature Assessment by Sliding Thresholds are valuable contenders for feature selection in the case of small sample size imbalanced data. In addition, a comparison was carried out by the researchers in regards feature selection and resampling methods for class imbalance: despite the fact that the experiments implied resampling would not enhance performance, they nonetheless supported the view that further work in this field remained necessary owing to the fact that he authors had carried out testing on only those combinations where feature selection was before resampling.

Very little emphasis was placed on the examination of the relevance of the application order of various different pre-processing methods. One corresponding work is that of [195]which involved examination of the combined effect of class imbalance and overlapping on classifier performance. A number of other researches have centred their attention on solutions to the co-occurrence of class imbalance and irrelevant features. An initial study [191]notably conducted

in relation to the Web categorisation field, made the statement that feature selection approaches are not always suitable in regards to imbalanced data sets. Accordingly, a feature selection model, which notably chooses features for both positive and negative classes on an individual basis, is suggested, with the subsequent features openly combined. One other study [192]expands on this through the adoption of feature subset selection prior to original dataset balancing with the aim of predicting the protein function from amino acid sequence features. In this case, the modified training set feeds a Support Vector Machine (SVM). In turn, this subsequently provides a greater degree of accuracy in the findings than those garnered when the same classifier underwent training through the original data. Regardless, however, a conflicting mix of approaches was not taken into account, meaning it is not possible to draw conclusions in regards to their suitability in relation to application order.

With the odd exception, there is very little research that presents a wide-ranging and in-depth comparison in regards the combined influence of resampling and feature selection on multi-class/binary learning. In an effort to fill this void, through the present work, resampling algorithm performance is experimentally examined when used in line with feature selection techniques for imbalance learning. In this case, the performance demonstrated by feature selection prior to resampling pipelines is compared, and vice versa, on KDD cup dataset, with the use of the 7 feature selection approaches, 3 widely implemented classifiers, and the resampling approach for class imbalance learning. The work's findings may deliver fundamental reference value for those professionals in the concept of data mining and machine learning, particularly when devising classification pipelines, therefore making recommendations in regards which are the most valuable and worthwhile combination to attempt and which should not be taken into consideration when seeking to overcome issues of imbalance learning.

3.5.1.4 Feature Selection for Sub-minor Attacks

In addition to that which has been discussed in this work so far, a number of different IDS have also been presented in other works. For example, in the study by[196], an IDS's performance, as based on the SVM, multivariate adaptive splines and linear genetic program, was assessed; this was done through the application of a novel significant feature selection algorithm, notably independent of the application of modelling tools. Notably, there is the removal of one input feature from the data; this is done one at a time. The dataset remaining after removal is then adopted in line with classifier training and testing. Subsequently, the performance of the classifier is then contrasted alongside that of the original classifier with specific consideration

to relevant performance criteria. Ultimately, there is the weighting of features in line with various rules centred on performance comparison.

In the work of [197], there was the in-combination application of the hidden Markov model and fuzzy logic with the aim of identifying intrusions. Through this method, the hidden Markov framework was applied in consideration to dimensionality reduction. In the study by [198], a wrapper-based feature selection algorithm was presented in mind of creating lightweight IDS. The modified RMHC was adopted as the search strategy whilst the evaluation criterion was outlined as the modified linear SVM. The method was found to achieve time efficiency in regards to the process of selecting features, with a high degree of detection achieved for IDS.

A minimal-Redundancy-Maximal-Relevance criterion (mRMR) was presented in the research carried out by[153], which focused on the selection of features on an incremental basis. Such a criterion outlines (mRMR) in mind of the incremental selection of features. Such a criterion provides the potential for features to be selected without incurring significant cost. The suggested method was contrasted alongside the maximal relevance criterion with the adoption of three different classifiers. The findings garnered provided validation that an mRMR feature selection has the potential to achieve classification accuracy enhancement.

A simple but nonetheless effective and time-efficient feature selection method was suggested by [154]in line with conditional mutual information(CMIM). The method presented was contrasted alongside other similar approaches, as in the case of C4.5 binary trees and fast correlation-based filter. When adopting this particular approach, there is a need for binary input features. It was determined that, alongside naïve Bayesian classifier, CMIM was seen to be more proficient than other approaches, as in the cases of boosting and support vector machine.

An IDS based on Flexible Neural Tree (FNT) was introduced through the study of [199], where the framework utilised a pre-processing feature selection stage in mind of achieving identification performance improvements. Through the use of the KDD Cup99, a 99.19% detection accuracy was achieved by the FNT model, utilising only 4 features. During more recent times, a forward feature-selection algorithm was presented by [133]through the application of the mutual information approach so as to complete measurement pertaining to the correlation between features. The optimal feature set was subsequently directed for the purpose of training the LS-SVM classifier and accordingly building the IDS. In the research carried out in [200], a SVM-based IDS—notably amalgamating a hierarchical clustering and

the SVM—was presented, with the hierarchical clustering algorithm applied in order to present the classifier with a greater degree of quality across training data with the aim of decreasing test time and average training so as to achieve classifier classification performance improvements. When measuring the SVM-based IDS, accuracy was seen to be 95.75%, with only a 0.7% false-positive rate, when utilising the corrected labels KDD Cup 99 dataset.

3.6 Summary and Conclusion

As has been established through the literature, both cost-sensitive and sampling methods have been commonly implemented. It was stated in the study by [98] that cost-sensitive learning was recognised as achieving greater levels of success; nonetheless, the point was posited that such approaches incorporate notable limitations owing to the fact they deal with an imbalance at the class level. In this regard, a greater degree of flexibility was seen to be achieved through sampling techniques, enabling various aspects of the dataset to be sampled and taken into explicit account the issue of minor disjoints. In this vein, the work of [201] concurs with the result of [102]; however, it remains that the literature findings show an ad-hoc approach as most commonly being adopted by researchers in this case. Weiss in[71] has provided several guidelines as to which methods can be recommended for dealing with specific problems with imbalanced datasets. However, current research indicates that the choice of sampling approach, and choice of distribution if not a random sampling, depends on the method and problem at hand as stated in [71], [102]. Similarly, for weighted approaches, determining optimal weights is also an ad-hoc process, which becomes increasingly complex the more classes there are.

In the research carried out in the case of class imbalance, very little attention has been directed towards multi-class issues [201]; however, a wide-ranging, in-depth work was carried out with the application of cost-sensitive ANNs, with the inclusion of 21 different datasets from the UCI repository. Oversampling effects were taken into account, alongside the effects of SMOTE sampling, under-sampling, and threshold moving for both ensembles and single classifiers. Despite the fact that the approaches were seen to be valuable in the case of two-class problems, it remains that multi-class problems did not benefit, with negative effects even witnessed in some cases. One explanation in this regard could be that there are additional classes, which could induce misclassification and accordingly increase problem complexity. One possible solution to this could be through changing the issue to the form of a number of different binary classification tasks and combining classifiers in line with each pair. Nonetheless, this would not be successful in changing weight-setting complexities; rather, as has been emphasised in the work of [91], this can actually heighten levels of complexity for users.

It is common for Feature Selection (FS) methods to be commonly categorised in relation to the interaction achieved between model construction and attribute selection processes. In regards filter approaches, it is most common for FS to be implemented on an independent basis, away from classifier design, with emphasis placed only on the more central factors of the features; when examining the wrapper approaches, in contrast, a feature subset is evaluated in consideration to the possible classification performance it could illustrate in the event it is adopted so as to create the classifier. Importantly, the relation between the feature and the class label is the only aspect taken into consideration by the filter model. When compared next to the wrapper model, lower computational costs are recognised. It is important to recognise that the assessment criterion is critical in this regard.

In line with the reviewed literature, two different methods of pre-processing are to be implemented in mind of solving the issue of imbalanced datasets and the misclassification of attacks, namely filtering feature selection and hybrid sampling. An explanation for the adoption of hybrid sampling is the need to circumvent useful data loss, which could potentially be removed through the application of the under-sampling approach; this could impact the learning process as a whole. Furthermore, over-sampling is not applied owing to the recognised possibility of the bias problem potentially increasing owing to the fact that an exact copy of the existing instance is made, meaning there will be an increase in over-fitting. In regards filter-supervised feature selection adoption, this will be implemented as, dissimilar to the wrapper methods; filter approaches are not seen to be comprehensive in terms of the ultimate learning algorithm. This degree of independence is acknowledged as one additional benefit linked with filter approach implementation.

Chapter 4 : Investigating the Machine Learning Classifier Behaviour in the Presence of Class Imbalance

4.1 Introduction

Recently in the areas of data-mining and machine-learning, dealing with applications that are subject to challenges of class imbalance has become recognised as an issue of significant importance. A number of different solutions have been introduced to overcoming the issue of class imbalance, including cost-sensitive learning, ensemble learning, and the use of data resampling as reviewed in Section 3.3. To the best knowledge of the author, these studies have not considered providing solutions to the multi-class categorisation of data in the presence of class imbalance, nor have any of the proposed approaches investigated solving the challenge based on rigorous empirical works. Therefore focusing on attaining improved classification performance in the presence of class imbalance, there is a need to consider the most effective and secure data classification approaches via rigorous experimentation. Those approaches that are most stable in relation to their ability to handle class imbalance should be found and recommended. In line with such a need, a rigorous empirical approach is carried out in this thesis in order to assess various performance criteria and a method's stability in the case of use within a network environment prone to intruder attack.

This chapter provides an overview of an empirical analysis centred on establishing the fundamental factors underpinning inadequate performance in the case of the majority of more widely known machine-learning classifiers, particularly in the case of learning from less significant/common attacks and classes. It is noted that the dataset of KDD Cup '99 that will be used in the experiments that will follow, which is also commonly used across studies in this field, is ultimately imbalanced in nature. This is owing to the fact that in the network intrusion application domain, some attacks are recognised as more recurrent/frequent, whereas some others are rare.

In line with the KDD Cup '99 dataset's amount of classes, addressing class imbalance can be viewed as solving a multi-class or binary class challenge. The existing studies in recommending solutions to the class imbalance issue within network IDSs approach the problem as a binary class classification problem due to the complexity associated considering the problem as multi-class classification. However such approaches are restrictive and will lead to time-consuming approaches requiring a complex collection of different classifiers to solve the underlying problem. Therefore across the study outlined in this thesis, the issue is viewed

as a multi-class classification problem. In line with this stance, this work explores the adoption of various machine-learning algorithms in an effort to circumvent the usual misclassification issues experienced by academics making use of the imbalanced KDD cup 99 datasets throughout the course of work. Suggestions are presented in line with the most highly recommended classifier for imbalanced data categorisation.

Section 4.2 presents the rigorous experimental setup used to gather data about the performance of classification frameworks to be explored. Section 4.3 analyse the results in detail leading to a summary and conclusions being provided in Section 4.4.

4.2 Experimental Setup

4.2.1 Data-Mining Tool

The development of Weka (Waikato Environment for Knowledge Analysis) toolkit for Data Mining was first documented at the University of Waikato, New Zealand [202]. Weka provides an experimental environment for the application of a large number of different and popular machine-learning algorithms. The machine learning algorithms are implemented within the toolkit and are readily available for use. The tool also provides effective data input, data pre-processing, post-processing, regression, visualisation, and analysis capabilities. Weka is defined as a Java software package presented via a GUI interface. A number of different benefits are garnered through the application of Weka, with the most noteworthy that of its free availability within the GUN public license. Furthermore, owing to the fact that its core utilises Java programming language and can, therefore, be executed across the majority of computing infrastructure, it is recognised as being convenient and manageable. Importantly, it is able to satisfy the main requirements in regards data-mining as a result of its approaches to data-processing and modelling.

4.2.2 Dataset and Pre-Processing

For the purpose of conducting the experiments presented in this, 10% of the original KDD Cup ’99 dataset is used (which is publically available for researchers). This reduction of the dataset was required as there were constraints in the memory and processing power of the computers used to support the experiments. Table 4-1 provides an overview of the number of records encompassed within the entire dataset in comparison to that utilised in the experiments conducted. It is noted that the 10% was a fair sample of the original dataset and did not significantly vary in terms of the presence of imbalanced data when compared to the original dataset.

Table 4-1: Number of records in Full KDD Cup '99 Dataset vs. 10% sample dataset used in experiments

Dataset Description	Num. of instances
Full KDD Cup '99 dataset	4,898,431
Selected 10% dataset of KDD Cup '99 dataset	494,020

Subsequently, the selected dataset's recognised duplicate records were taken out of the data under examination in order to further decrease computational time-consumption and memory, whilst also facilitating the avoiding of the bias towards particular classes. As has been stated in the study carried out by [71], following the removal of duplicate files, an increase in the identification and categorisation of minor classes was witnessed through the adoption of ANN and DT algorithms. Table 4-2 provides an insight into the number of instances present in the selected 10% sample set both with and without its duplicate records.

Table 4-2: Number of records in 10% sample dataset with and without duplicates

Dataset Description	Num. of instances
The sample 10% of the KDD Cup '99 dataset with duplicates.	494,020
The sample 10% of the KDD Cup '99 dataset with duplicates removed.	145,584

4.2.3 Validation Methods

In presenting the validation results of the performance of classifiers, there are two key methods that can be used, namely the percentage split, and cross-validation. In line with the completion of the validation process, it is notable to state that there is a differentiation in the findings owing to the fact that choosing the split of test and training data from within a given sample dataset is a sensitive task. Generally, the size of the training-testing dataset split has an impact on the overall classifier accuracy. However different classifiers will perform differently to the percentage split values.

Table 4-3 provides the classification accuracy results when different classifiers are used alongside holdout validations of 60%, 80%, and 90% respectively. An N% holdout validation refers to N% of a dataset been used for training and (100-N)% been used for testing. The classifiers compared in the experiments include J48, Random forest (RF), Bayes Net (BN) and Naïve Bayes (NB). When examining the classification accuracy of various classifiers across all attacks, there appear many variations, which is seen to be influenced sometimes by the test set containing new types of attacks that potentially have not been included in the training set.

One noticeable contrasting result with this observation is the J48 classifier which is able to only identify 36.4% of U2R attacks when there is a 90% holdout, although this notably increases to 50% and 56.6% respectively when the % split of training data is decreased to 80% and 60%, respectively. It is believed that this observation is due to the rather very low test dataset used in testing when a 90% holdout is used and one wrong classification during testing will result in a very large reduction in the overall accuracy figure. The above observations show that despite the fact that some classifier identifications demonstrate slight shifts, accuracy as a whole is nonetheless influenced by the percentage split in training/testing data. In contrast in the case of using NB classifier, as an example, there is a drop in the accuracy as a whole, notably from 98.7% to 89.9 %, when there has been a reduction in the training set from 90% to 80%. Figure 4-1 coupled with Table 4-3, provide a more in-depth overview of these results.

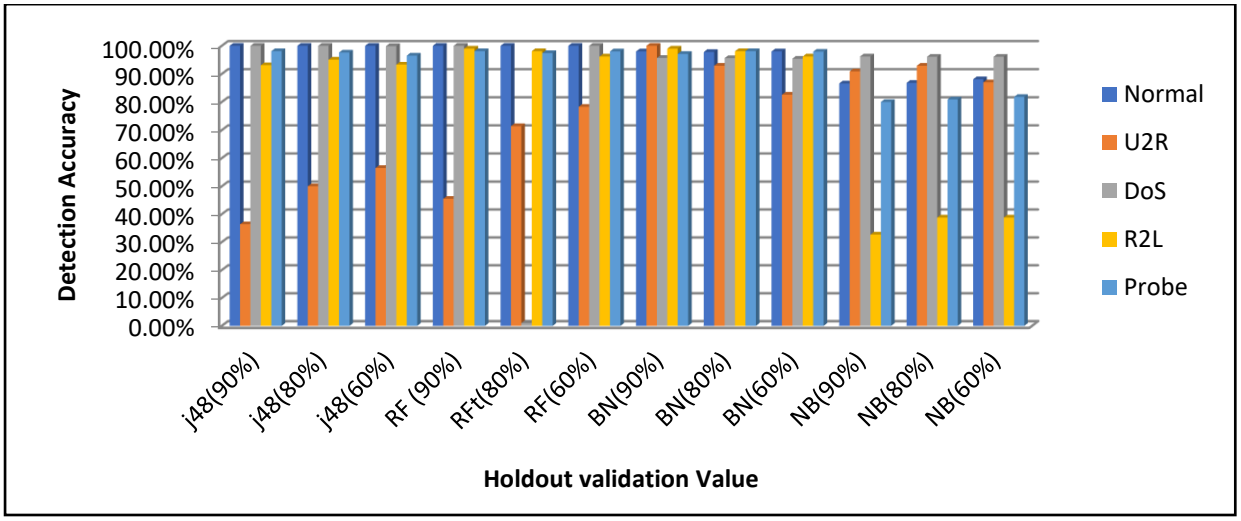


Figure 4-1: Classifiers' performance with different holdout validation value

As can be seen from the non-conclusive results detailed in Figure 4-1 about the variation of the accuracy of different classifiers used in attack classification, with the percentage split of training/testing data, a statement can be made that a greater degree of reliability of accuracy Figures can be achieved through the use of N-fold cross-validation. Importantly, the most common value of N, in the view of [203] is tenfold cross-validation; therefore this value for N is selected in this thesis owing to the fact that the empirical work with DTs and NB, are both examined and supported by [204].

Table 4-3: Classifiers' performance with different holdout validation values

	Overall Accuracy(weighted Average)				Accuracy by Classes				
Percentage %	**Accuracy**	**TPR**	**FPR**	**precision**	**Normal**	**U2R**	**DoS**	**R2L**	**Probe**
Percentage of 90%									
J48	99.9	99.9%	0.2%	99.9%	100%	36.4%	100%	93.1%	98.1%
Random Forest	99.91	99.9%	0.1%	99.9%	100%	45.5%	100%	99.0%	98.1%
Bayes-Net	97.14	97.1%	0.6%	98.6%	98%	100.0%	96%	99.0%	97.1%
Naïve-Bayes	98.7	98.9%	1.9%	96.8%	86.6%	90.9%	96.2%	32.7%	80.0%
Percentage of 80%									
j48	99.9	99.9%	0.1%	99.9%	100.0%	50.0%	100.0%	95.1%	97.6%
Random Forest	99.9	99.9%	0.1%	99.9%	100.0%	71.4%	1.0%	98.1%	97.4%
Bayes-Net	97	97.0%	0.8%	98.5%	97.8%	92.9%	95.6%	98.1%	98.1%
Naïve-Bayes	89.9	89.9%	1.9%	96.9%	86.8%	92.9%	96.1%	38.8%	80.9%
Percentage of 60%									
j48	99.8	99.8%	0.2%	99.8%	100.0%	56.5%	99.9%	93.3%	96.5%
Random Forest	99.9	99.9%	0.1%	99.9%	100.0%	78.3%	100.0%	96.2%	98.0%
Bayes-Net	96.98	97.0%	0.8%	98.5%	98.0%	82.6%	95.4%	96.2%	97.9%
Naïve-Bayes	90.6	90.6%	1.9%	96.8%	88.1%	87.0%	96.1%	38.8%	81.8%

4.2.4 The Classifiers and Assessment Metrics

In line with literature that focused on evaluating imbalanced datasets, with regards to the bias inherent in various classifiers towards to the major class/s, it is noted that the use of a number of popular classification methods has been investigated. These classifiers fall under the tree-based classification approaches, e.g. in particular J48, Random Forest and Bayes Network classification approaches, for example, NB and BN. However such works have not been rigorous and have failed to make viable conclusions.

Evaluating classifiers' performance will be centred on the IDS's overall assessment metrics, as highlighted in Section 2.4.1. Importantly, the general overall accuracy of the classification of an imbalance dataset by a given classifier will not highlight the classifier's ability to identify different classes of attacks owing to the fact that they do not differentiate between the accuracy of classification of different classes in an imbalanced dataset, having significantly varied amount of instances between classes. Accordingly, this could potentially result in inaccurate conclusions being drawn, such as in the case of a classifier attaining 90% overall accuracy across a large dataset but with a rather low accuracy of only 10% demonstrated in the classification of a minor attack class such as R2L. Accordingly, the general overall accuracy for the entire database being investigated, the accuracy of classification of each type of class, precision, true positive rate, and false-negative rate are all used as assessment metrics in the research conducted within the scope of this thesis.

4.3 Investigating the Classification of Imbalanced Datasets

In this section, the experiments are conducted for the classification of an imbalanced dataset using ensemble learning approaches, e.g. cost-sensitive, ensemble learning, and sampling algorithms.

4.3.1 Classifiers' Performance after Resampling

As has been given due consideration in Section 3.3.1, this research takes the view that 're-sampling' is the most widely used approach to improving classifier performance, in the presence of a data imbalance to avoid undue bias towards major classes and erroneous classification of, in particular, minor classes. In the research conducted in this thesis, the sampling value was varied from 0-1 in steps of 0.1 (see Figure 4-2), with the increase of sampling value influencing a decrease of detection of instances of the major classes (reduction accuracy) e.g. Normal and DoS; however indicating a slight increase in the detection of minor classes such as U2R and R2L. Notably, Figure 4-2 and Table 4-4 provide an overview of the detection accuracy and distribution of a number of instances within a class, with respect to the sampling values used. Results tabulated in table 4.4 indicate the balancing of various kinds of attacks when the sampling rate is increased from 0.0 to 1.0 in steps of 0.1. As the steps below 0.1 lead to smaller value/instances for the minor classes that let the dataset to still be imbalanced, see Table 4-5 presents some of them and how far the instances of each class changed.

Figure 4-2: Classes distribution with different sampling value (with Percentage)

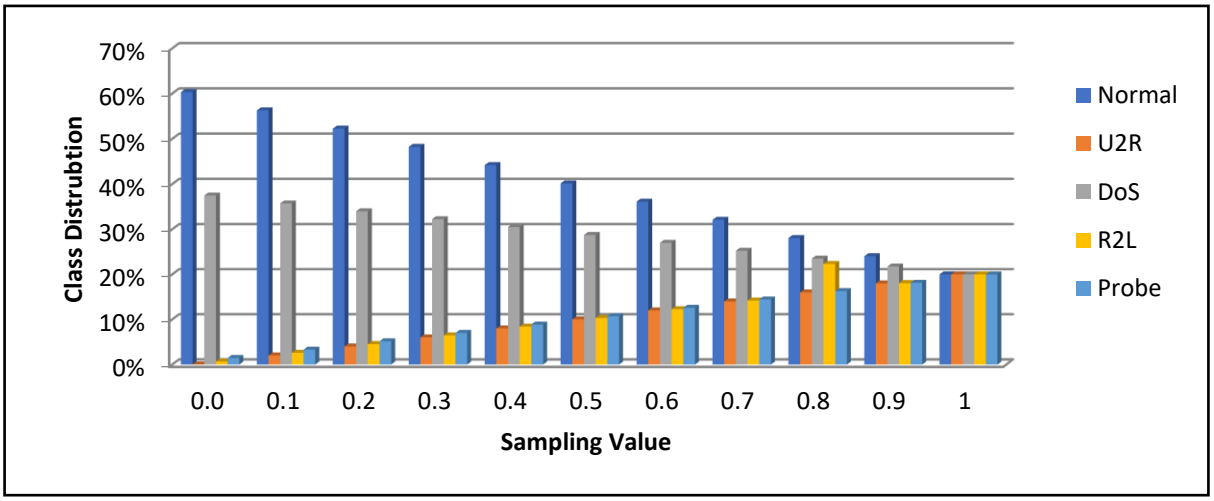


Table 4-4: Classes distribution with different sampling value (with value)

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Normal	87831	81959	76088	70216	64354	58473	52602	46731	40859	34988	29116
U2R	52	2958	5864	8771	11677	14584	17490	20397	23303	26210	29116
DoS	54572	52026	49480	46935	44389	41844	39298	36753	34207	31662	29116
R2L	999	3810	6622	9434	12246	15057	17869	20681	32493	26305	29116
Probe	2130	4828	7527	10226	12924	15623	18322	21020	23719	26418	29116

Table 4-5: Classes distribution with different sampling value below 0.1

	0.0	0.1	0.01	0.02	0.001	0.05
Normal	87831	81959	87243	86656	84895	87772
U2R	52	2958	342	633	1505	81
DoS	54572	52026	54317	64062	53299	54546
R2L	999	3810	1280	1561	2404	1027
Probe	2130	4828	2399	2669	3479	2156

As can be seen, when reviewing Table 4-4, all classifiers taken into examination have high overall accuracy. With regards to the categorisation of an attack, it is seen that the rate of identification across major classes is high; nonetheless, identification classification remains low for some classes owing to imbalanced data distribution. On a global scale, the rate of identification with regards the examined categorisation of classes is influenced by the imbalanced dataset, where the major classes do receive some degree of bias. As an example, 99% accuracy is achieved by the RF classifier performance in the categorisation of major classes, although the accuracy of classification is only 67.3% of for U2R attacks and 67.9% for R2L attacks. Furthermore, for BN the overall accuracy is recognised as 97.2%, with the identification rate across all classes been notably high, but with the exception of the minor class U2R which indicates an accuracy of only 82.7%.

Table 4-6: Classifiers performance before resampling

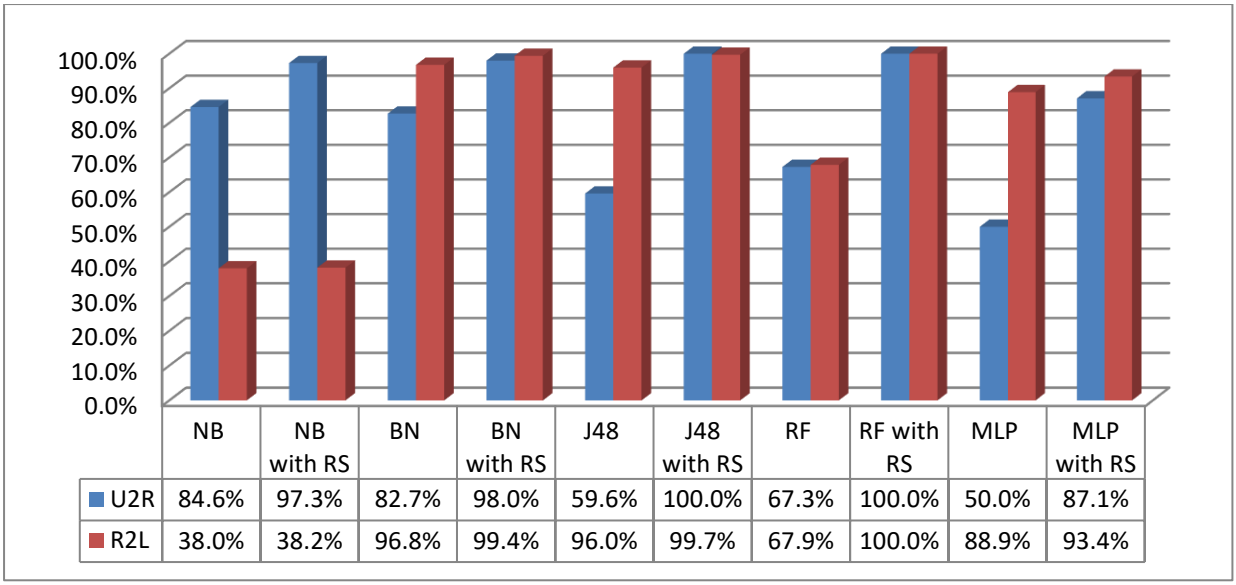
	Overall Accuracy(weighted average)				Accuracy by Class				
	Accuracy	**TPR**	**FPR**	**Precision**	**Normal**	**U2R**	**DoS**	**R2L**	**Probe**
NB	89.8	89.8%	1.8%	96.9%	86.6%	84.6%	96.1%	38.0%	81.3%
BN	97.2	97.2%	0.7%	98.6%	98.1%	82.7%	95.8%	96.8%	98.1%
J48	99.8	99.0%	0.1%	99.9%	99.9%	59.6%	100.0%	96.0%	98.2%
RF	99.94	99.9%	0.1%	99.9%	99.9%	67.3%	100.0%	67.9%	98.8%
MLP	99.9	99.7%	0.3%	99.7%	99.9%	50.0%	99.7%	88.9%	98.4%

Table 4-7: Classifiers performance after resampling

	Overall Accuracy(weighted average)				Accuracy by Class				
	Accuracy	**TPR**	**FPR**	**Precision**	**Normal**	**U2R**	**DoS**	**R2L**	**Probe**
NB with resample	87.9	87.9%	1.9%	94.1%	85.0%	97.3%	96.3%	38.2%	81.3%
BN with resample	97.4	97.4%	1.2%	97.8%	98.2%	98.0%	96.0%	99.4%	98.4%
J48 with resample	0.999	100.0%	0.1%	99.9%	99.9%	100.0%	100.0%	99.7%	99.7%
RF with resample	99.98	100.0%	0.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.0%
MLP with resample	99.34	99.3%	0.4%	99.3%	99.7%	87.1%	99.9%	93.4%	98.4%

Owing to the fact that one approach to resolving the challenge of classification of multi-class imbalanced datasets is that of using data sampling, Table 4-6 provides an overview of the accuracy of classification of various classes under examination in this thesis and their detection accuracies following the application of data resampling to the dataset. As can be seen, the identification rate of the minor classes demonstrates an improvement, with the major classes not being recognised with any significant bias. For example, the minor class, U2R's identification accuracy has increased from 59.9% to 100% when adopting the j48 classifier. Nevertheless, when using the NB classifier, the identification accuracy of R2L still remains rather low at 38%. In chapter 5 we show that this challenge can be easily overcome by investigating data imbalance within the R2L class.

Figure 4-3 provides a comparison of classifiers' performance in the presence of data imbalance both prior to resampling and following the application of data resampling.



	NB	NB with RS	BN	BN with RS	J48	J48 with RS	RF	RF with RS	MLP	MLP with RS
U2R	84.6%	97.3%	82.7%	98.0%	59.6%	100.0%	67.3%	100.0%	50.0%	87.1%
R2L	38.0%	38.2%	96.8%	99.4%	96.0%	99.7%	67.9%	100.0%	88.9%	93.4%

Figure 4-3: Classifier performance when data resampling is applied

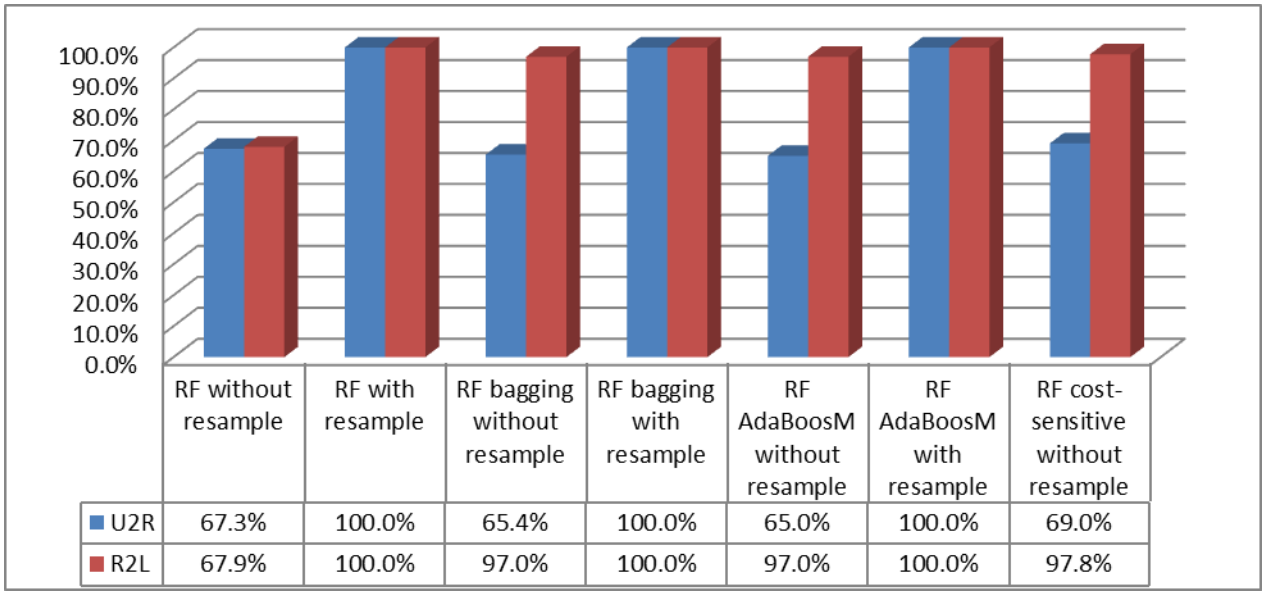
4.3.2 Random Forest Behaviour in the Presence of Imbalanced Data

This particular experiment centres on the detailed investigating of the RF classifier's performance in the presence of imbalanced data. It seeks to establish the most appropriate and valuable learning approaches that can be adopted in striving to improve the minor classes' accuracy of classification when using the RF classifier. The learning techniques investigated in this case include cost-sensitive, ensemble, and resampling.

The experimental results are tabulated in Table 4-7. Figure 4-4 further highlights the findings pertaining to the use of the different learning approaches with RF as the main/base classifier. The experimental results indicate that minor attacks, such as U2R and R2L are correctly categorised by RF without any instance of misclassification, achieving an overall accuracy level of 99.98, with most types of attacks classified at 100% accuracy after using data resampling. Particular attention is given to the investigation of performance accuracy shown by the use of the cost-sensitive learning approach suggested by [109]. It is recognised that this particular classifier has a notable impact on RF classifier's performance behaviour with regards to the classification accuracy of the minor class, R2L owing to its significant 30% increase. On the other hand, the identification rate for the minor class U2R has only been impacted to a minor degree, with correct classification amounting to 69% only marginally higher than when compared to using a pure RF classifier. The point is argued that the RF algorithm functions in a manner similar to bagging as it is seen to be a form of ensemble learning. However, as can be seen, when reviewing the results, it is apparent that RF with bagging classifier demonstrates

a better degree of performance than RF as a tree base, pure classifier. However, more importantly, the rate of identification of minor class R2L has increased from 67.9% to 97%.

A further intention of the experiment carried out in this section was to examine RF classifiers performance when the ensemble learning approaches referred to as AdaBoosM [205]and Bagging [206] are used. It is shown that RF classifier performance with AdaBoostM is the same as when using bagging; an increase in the R2L identification rate was witnessed in both cases, whereas a decline of accuracy was witnessed in the case of classification of the attack U2R when data resampling was not carried out. It is noted that this particular experiment has adopted two-hybrid imbalance learning methods, namely bagging with resampling and AdaBoostM with resampling. As shown by the findings, the ensemble methods were positively affected by data resampling, with all minor categorisations, in addition to major categorisations, correctly classified. The conclusion may, therefore, be drawn that, when completing multi-classification using RF, data resampling is highly recommended.



	RF without resample	RF with resample	RF bagging without resample	RF bagging with resample	RF AdaBoosM without resample	RF AdaBoosM with resample	RF cost-sensitive without resample
U2R	67.3%	100.0%	65.4%	100.0%	65.0%	100.0%	69.0%
R2L	67.9%	100.0%	97.0%	100.0%	97.0%	100.0%	97.8%

Figure 4-4: Random Forest performance when using different learning methods and data resampling

Table 4-8: RF performance when using different learning method

	Overall Accuracy(weighted average)				Accuracy by Class				
	Accuracy	TPR	FPR	Precision	Normal	U2R	DoS	R2L	Probe
RF without resample	99.94%	99.9%	0.1%	99.9%	99.9%	67.3%	100%	67.9%	98.8%
RF with resample	99.98%	100%	0%	100 %	100%	100%	100%	100%	99%
RF bagging without resample	99.93%	99.9%	0.1%	99.9%	100%	65.4%	100%	97%	98.7%
RF bagging with resample	99.98%	100.0%	0.0%	100%	100%	100%	100%	100%	99 %
RF AdaBoosM without resample	99.93%	99 %	0.1%	99.9%	100%	65%	100%	97%	98.6%
RF AdaBoosM with resample	99.98%	100 %	0 %	100%	100%	100 %	100%	100 %	99.9%
RF cost-sensitive without Resample	99.95%	99.9%	0.1%	99.9%	100%	69.0%	100%	97.8%	99.2%

4.3.3 Classification of Minor Attacks with the Naive Bayes Classifier

Experiments carried out in this section revealed that data resampling affects the NB classifier behaviour, as indicated by the increase in identification ability for U2R, whereas the R2L classification accuracy remains the same (see Figure 4-3). Accordingly, a number of experiments have been carried out in order to establish reasons for this behaviour and to propose alternative approaches that will enable NB classifier to identify R2L attacks accurately. Therefore experiments are carried out to investigate the performance of a number of methods that can be used to classify imbalanced datasets, i.e. learning approaches, namely bagging, cost-sensitive and stacking.

Results in Figure 4-5 show that in the case of R2L attacks, identification accuracy has not been changed through the application of the cost-sensitive approach, even when the R2L weight classification (see Section 3.3.2) was increased beyond 2. Furthermore, when the popular 'bagging' ensemble learning approach is applied in this regard with NB as the base classifier, a slight increase in accuracy of detection in the case of the U2R class (increasing from 84.6% to 86.6%) is reported, whilst for R2L the accuracy approximately remains the same. When bagging is used with data resampling, there is a further increase of accuracy in U2R identification (98.3%), whereas R2L is not seen to demonstrate any change in detection accuracy (approximately no change at 38.5%).

One further ensemble method is that referred to as stacking, which is recognised as bringing together various classifiers, with NB selected as the base classifier. When carrying out the first of the experiments, the use of NB alongside bagging was applied, with RF selected as the base classifier within bagging. The initial accuracy results were recorded without resampling, which demonstrated results with an indication of a clear decline in the identification rate for U2R (by 50%), as well as a high increase in the identification rate of R2L (by 52%). Nonetheless, upon completing the experiment again later, this was carried out with resampling, achieving a positive outcome: the minor class was correctly classified, with an increased accuracy score achieved at 100% for U2R and 98.5% for R2L. Other methods were also examined, the MLP classifier being one of them, which was applied as the base classifier for bagging, with NB used as the base classifier of the stacking stage; the performance of NB with regards to the attack R2L identification did not change. However, U2R attach detection was completely disregarded with 0% accuracy. In another case, NB provided the base of stacking, whilst for bagging the base was J48; NP performance was seen to demonstrate a decrease in both R2L and U2R classification.

The conclusion is drawn that, in the case of R2L, NB classification is improved through the adoption of a hybrid method bringing together elements of the data-level approach and those of the ensemble approach, with the NB and RF bagging method seen to be the most highly recommended of those implemented.

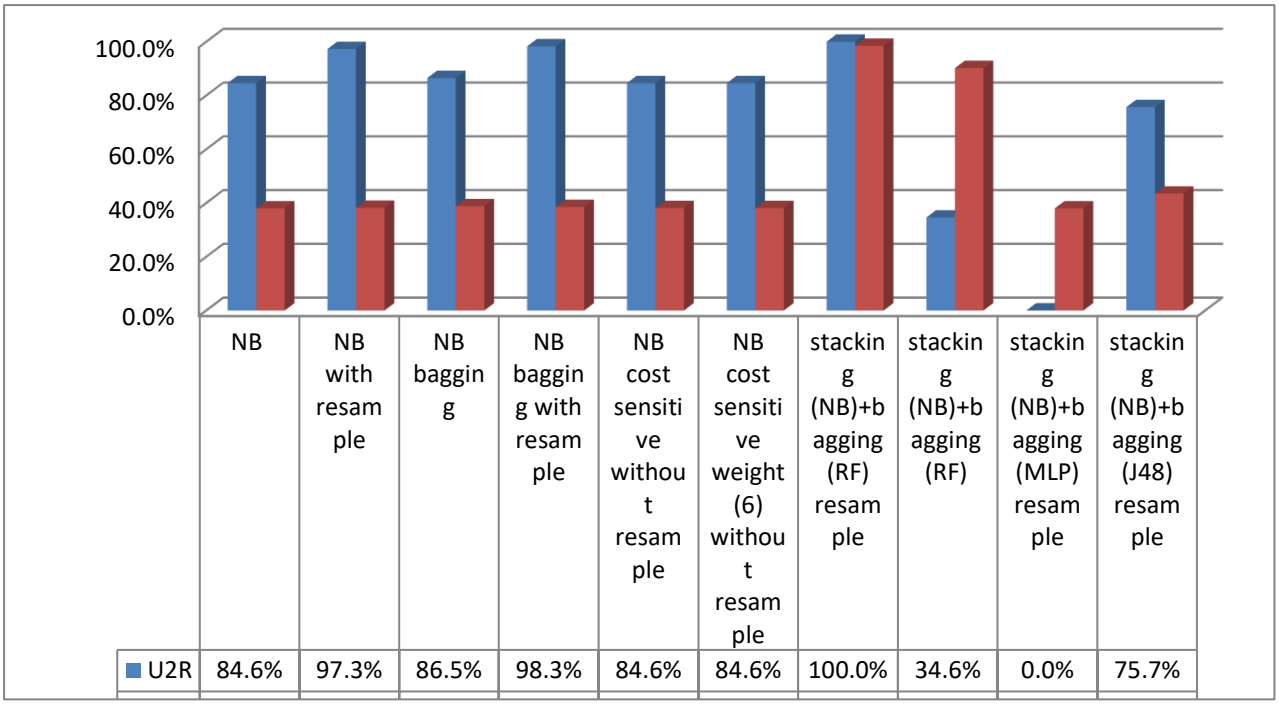


Figure 4-5: NB based detection of U2R and R2L attacks

4.4 Summary and Conclusion

The research presented in this chapter has directed its attention towards the challenge of multi-class classification in an imbalanced network IDS dataset, i.e. the widely used KDD Cup '99 dataset. A detailed analysis of using different machine learning algorithms popularly used to classify data was carried out. Following this in-depth exploration, a number of classifiers were found to be biased towards the major classes; subsequently, resulting in the minor classes being incorrectly classified.

The conclusions of the research conducted in this chapter are summarised as follows:

- There is a strong recommendation for the use of cross-validation as against the use of hold-out validation in the performance analysis of a classifier.
- The bias towards major classes, coupled with the poor performance of some classifiers, is a result of data imbalance between classes. It was shown that this can be overcome via the application of data resampling prior to classification.
- The Random Forest classifier's performance as an ensemble learning approach demonstrates its improved outcomes when compared with the use of traditional tree-based classification algorithms. This is due to the fact that RF utilises a subset of features in order for dividing tree nodes throughout the course of application of bagging, taking into account all aspects for splitting a node. Accordingly, the conclusion is drawn that the distribution of dataset elements between classes and the imbalance that exists influences RF performance. Accordingly, when using RF classifiers, it is recommended that data-resampling is applied within ensemble methods, namely AdaBoosM or bagging.
- For classification of the minor attack R2L, NB classification demonstrates improved outcomes via a hybrid method that takes and brings together the data-level approach and combines this with an ensemble approach. In this particular work, the recommendation is to apply the NB method alongside RF bagging.

In line with the above observations and conclusions, there is a recognised need to conduct further R&D work examining the factors that underpin the R2L attack's misclassification, even when using cost-sensitive learning and resampling. Chapter 5 will seek to explore this research challenge in more detail.

Chapter 5 : Class Imbalance within Class for the Minor Attacks

5.1 Introduction

In the field of machine learning, one of the most widely recognised fundamental challenges is the presence of imbalanced data, where such datasets are seen to relate to those data with a skewed class distribution. In such instances, at least one of the classes is seen to have a much larger number of samples (an example of a major class) when compared with the others (i.e. especially w.r.t minor classes). The issue with imbalanced datasets is that the classical machine learning algorithms (e.g. ANN and DT) are biased towards the major class (es), as highlighted in the study of [107]. Therefore, a poor classification rate is witnessed amongst the minor classes. In this case, at times, some classifiers opt to disregard the minor classes, as has been investigated in various works[73], [84], whereas a significant overall accuracy is still achieved as proven by the work of [71]. Due to this latter reason in many research work presented in literature the classifiers are operated with the assumption that there is a balance in the dataset's class distribution. However, such an approach will not lead to an optimal classification of data in the presence of class imbalance.

In Chapter 4 the issue of class imbalance was investigated with respect to network attacks. Conclusions were made as to how one could deal with optimising the accuracy of detecting both major and minor network attacks within an IDS. However, this work also concluded that some of the attacks do demonstrate class imbalance, within-class, due to the presence data skewness as a result of attacks that can be classified as sub-minor attacks within minor attacks or sub-major attacks within major classes. As in major classes, the number of data instances is high, our detailed analyses revealed that sub-major classes do not have a negative impact of the classification of major attacks [71], [207]. On the contrary, sub-minor attacks have a significant impact on the classification accuracy of minor classes as will be investigated in this chapter. Hence this chapter is dedicated to the study of optimising sub-minor attack classification in network IDS.

For clarity of presentation, this chapter is divided into several subsections. Apart from this section which provides an introduction to the research problem to be investigated in this chapter, Section 5.2 defines within-class data imbalance within the context of imbalanced datasets. Section 5.3 the well-structured research methodology adopted in this chapter to investigate within-class data imbalance. Section 5.4 provides an experimental system design flowchart for clarity of presentation. Section 5.5 provides experiment setup, and in Section 5.6,

NB detection accuracy across minor attacks are discussed. Section 5.7 anlayse the Factors Underpinning Inadequate Identification of R2L by NB. Section 5.8 presents summary and discussion.

5.2 Class Imbalance within a Class

The class imbalance within a class relates to the small disjuncts in data, which are recognised as been the data sub-groups, within a class. In this vein, the investigations carried out in [208] concluded that these small disjuncts resulted in producing classification errors. The authors in [207].presented that in some occasions the classifiers ignore the minor disjuncts, as the larger disjuncts are used to build the model. Accordingly, conclusions were made that class imbalance between classes might not be able to achieve optimal classification performance when data imbalance exists within some classes. A number of attempts have been made since to address the issue of both between class and within-class imbalance. Furthermore, it was established in [71] that specifically within-class imbalance within minor subclasses (i.e. due to the presence of sub-minor classes) can subsequently result in poor classification accuracy between classes (major vs minor).

5.3 Research Methodology

5.3.1 The Proposed ML Framework to Address within Class Imbalance

The proposed framework has been developed in line with the internationally accepted framework for data mining, CRISP-DM (Cross-Industry Process for Data Mining)[209]. This framework is widely adopted in order to overcome challenges with the adoption of DM [210] in practical application domains. It consists of progressive stages of design considerations and models, as illustrated in Figure 5-1 and summarised below:

1. **Business insight**: In the event of classification of minor attacks, poor accuracy of detection and hence performance was witnessed during investigations carried out in this thesis, when using the Naïves Bayes classifier, used widely for the purpose of attack classification in existing IDS.
2. **Data insight**: Through the use of the KDD ’99 dataset for experiments, the presences of a number of duplicate records were established, within and between minor classes. The dataset was also found to be imbalanced (see Chapter 3).
3. **Data preparation**: Data pre-processing is carried out, ensuring that major class bias and skewed class distribution are both circumvented through dataset resampling and duplicate records removal. [See Section 5.5 ‘experimental setup’ for more details].

4. **Modelling**: In this chapter, we apply different ensemble learning techniques and algorithms to determine the best model that enhances the detection accuracy of minor attacks when the most frequently used classifier in-network IDS, i.e., NB is used. [See section 5.5 for more details].
5. **Evaluation**: Evaluating each technique investigated with the use of attack/class detection accuracy metric to verify that business/application goals are met and to confirm the best model for the given application. [See Section 5.6].
6. **Application**: Information related to data collection, modelling, and implementation of the proposed framework is included in this thesis, enabling knowledge exchange with the R&D community in network IDS enabling the framework's practical application.

5.3.1.1 Business Insight

The majority of investigations carried out in the network IDS domain have sought to establish the most effective framework for attack classification with the aim of improving and enhancing the overall attack classification accuracy. In such attempts, it has been assumed that the classes are balanced. Appreciating the possibility of class imbalance, some authors have investigated the use of ensemble learning algorithms with various base classifiers; however, the detection accuracies of all type of attacks have not been documented in detail. In such work, the attention was only directed towards various classifiers' overall performance. In the presence of class imbalance, the proposed algorithms were therefore noted to encompass bias in relation to the major classes, which therefore results in a significant false error rate particularly in the case of the detection of minor classes.

However, a smaller number of different studies have focused on examining attack detection in imbalanced datasets, with such techniques only investigating class imbalance, between classes (i.e. major vs minor). To the best of the author's knowledge, no work has been carried out in studying the challenges faced by the presence of sub-minor attacks, i.e. the presence of within-class data imbalance in minor classes. For example in the case of the KDD '99 dataset, the sub-minor classes R2L and U2R (sub-minor attacks) are present but no study has investigated in detail whether the presence of sub-minor attacks is the reason for the reduced accuracy often demonstrated in detecting minor classes. This is the key focus of the research presented in this chapter.

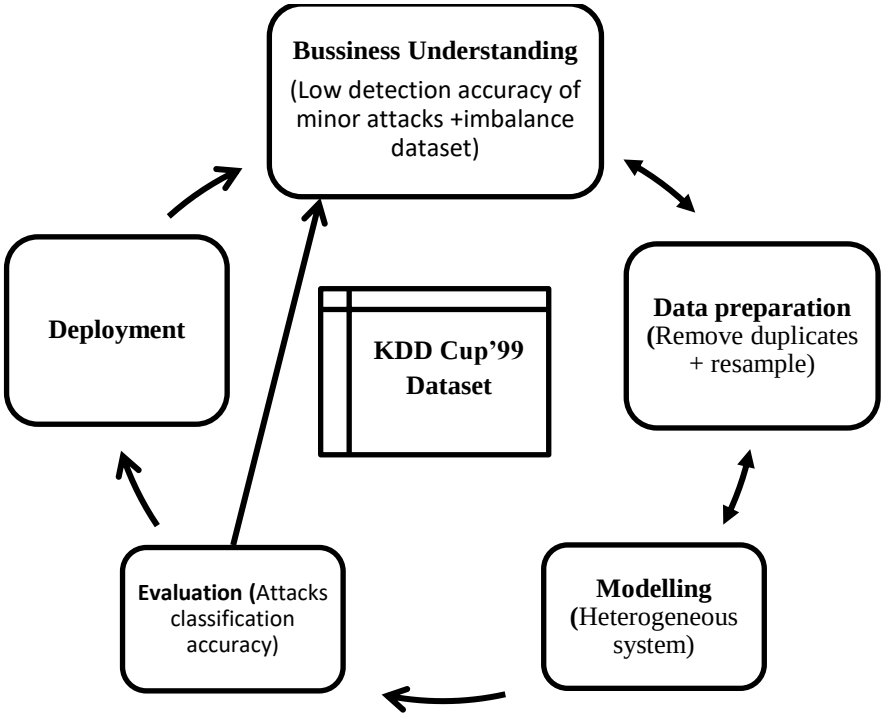


Figure 5-1: Proposed imbalanced dataset classification methodology

5.3.1.2 Data Insight

When utilising the KDD Cup '99 dataset, a critical analysis was carried out as presented in Section 3.2. The most pressing of considerations warranting research attention was shown to be the imbalanced nature of the dataset and the presence of duplicate records. In specific regards to the classes of this dataset, there is a total of five classes, each of which was shown to have sub-classes. The dataset classes, in addition to sub-classes, are shown in Table 5.1.

5.3.1.3 Modelling

Across the course of this research work, the application of a number of different ensemble methods and approaches will be examined, and the results will be analysed in detail. The proposed investigation framework is illustrated in detail in the Figure 5-2. In this case, Meta-learning (here stacking is used as described in section 3.3.3) provides the basis of the proposed heterogeneous system. The suggested framework comprises of two different phases: in the case of the first, there is the transfer of the pre-processed/balanced training dataset to the Navies Bayes algorithm, and the decision boundary of the class is studied for correct classification. Accordingly, NB output is adopted as the input to a Random Forest algorithm, which is viewed as being a second-level meta classifier. Importantly, the classes' decision boundary which is identified as being close to the neighbour class is entirely circumvented. More details of this model will be provided in section 5.4.

5.3.1.4 Evaluation

Throughout the evaluation phase, the suggested framework undergoes an assessment and a detailed analysis will be carried out. The metric applied for assessment is the attack accuracy. Owing to the fact that the emphasis of this work is centred on minor attacks, the accuracies of the detection of these attacks are detailed. Accordingly, when examining the overall accuracy of minor attacks, it is seen to be affected by the accuracy of their sub-minor attacks (R2L and U2R).

5.4 The Heterogeneous Model

In this chapter, the proposed classification approaches are structured as heterogeneous systems, which are predominantly based on stacking, as discussed earlier in the work. Figure 5-2 details the framework's adopted workflow, with the preliminary stage concerned with pre-processing, meaning there is the creation of the balanced dataset through the adoption of imbalance learning within classes. Following, there is the building of a heterogeneous system, which is focused on improving NB detection for minor attacks. The system is seen to comprise two different stages: the first stage considers the passing of the training dataset through to the NB classifier, with the sub-minor classes providing the basis for the NB to build classification. Accordingly, following this stage, NB related performances and results will be utilised as an input for bagging with RF as the base classifier. Therefore, based on the decision that is agreed on the heterogeneous system the classification result will be produced. Accordingly, there is the adoption of the evaluation metric so as to assess the generated decision. In this case, the detection accuracy for all minor attacks is considered as the evaluation metric.

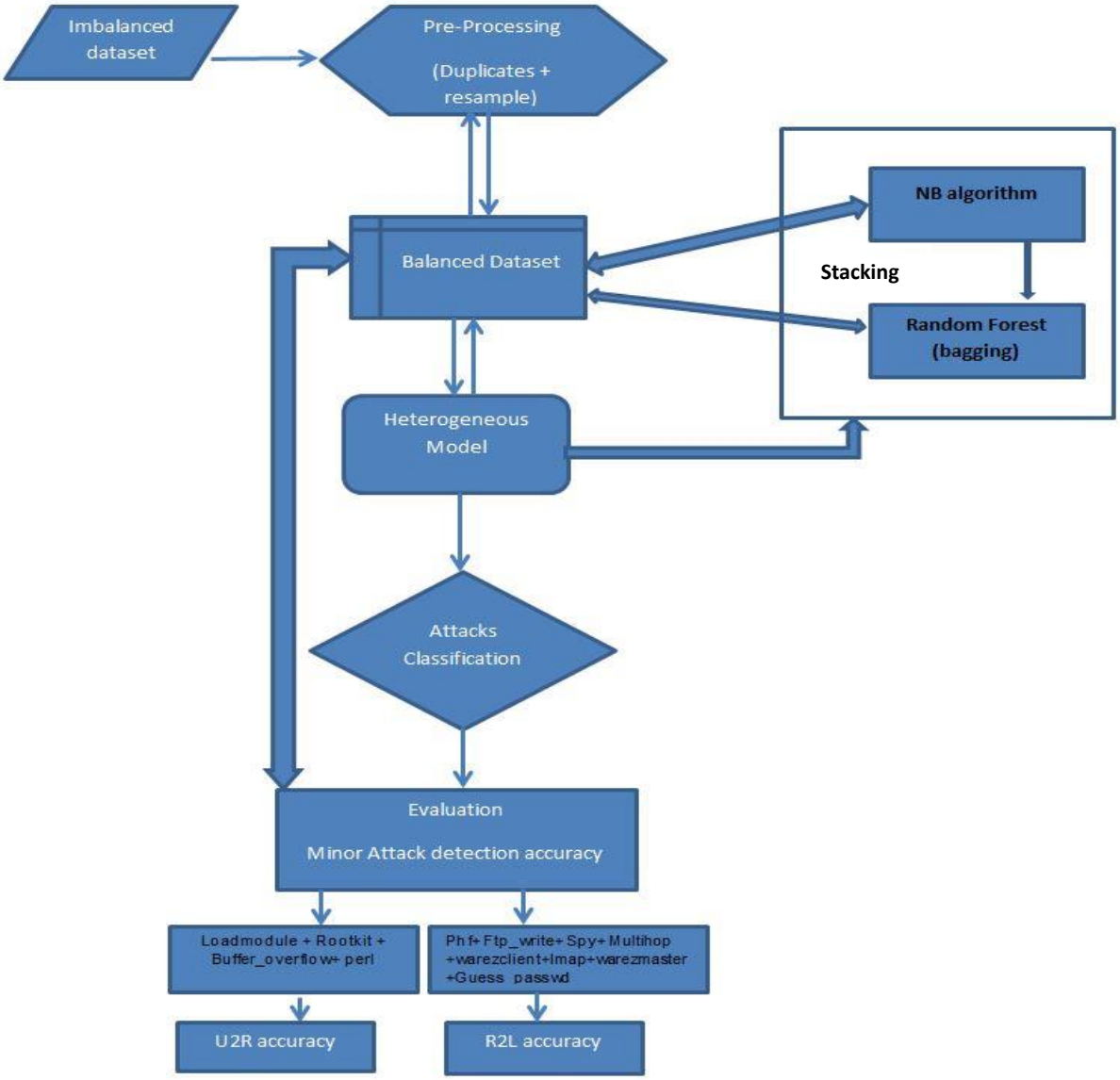


Figure 5-2: Flowchart for the study methodology

5.5 Experimental Setup

In an effort to achieve dataset balance and to further circumvent skewed class distributions; two dataset pre-processing stages are implemented: the removal of duplicate records and resampling. Notably, in this work, the sampling value taken is 0.1, as the other values (over 0.1) were experimentally determined to have an on the various instances of the major class. With this value, however, there is a minor increase in the minor class, whilst a slight decrease is seen in a major class. Neither of these changes was found to have an impact on the dataset's original distribution. Table 5-1 provides the count of sample values, of each class before and after each of the two pre-processing stages. It shows that classes with U2R, R2L categorisation has a relatively lower amount of samples, before sampling when compared to classes with categorisation as DOS, Probe, etc.

Table 5-1: KDD cup dataset class distribution before/after pre-processing and resampling

No	Attack Class Label	Count with Duplicates	Count without Duplicates	Count without Duplicates and with resampling	Attack Category
1.	back	2203	968	1504	DoS
2.	Teardrop	979	918	1459	DoS
3.	Loadmodule	9	9	641	U2R
4.	Neptune	107201	51820	47270	DoS
5.	Rootkit	10	10	641	U2R
6.	Phf	4	4	636	R2L
7.	Satan	1589	906	1448	Probe
8.	Buffer_overflow	30	30	659	U2R
9.	ftp_write	8	8	640	R2L
10.	Land	21	19	650	Dos
11.	Spy	2	2	634	R2L
12.	Ipsweep	1247	651	1218	Probe
13.	Multihop	7	7	639	R2L
14.	Smurf	280790	641	1209	DoS
15.	Pod	264	206	818	DoS
16.	Perl	3	3	635	U2R
17.	Warezclient	1020	893	1436	R2L
18.	Nmap	231	158	775	Probe
19.	Imap	12	12	643	R2L
20.	Warezmaster	20	20	650	R2L
21.	Portsweep	1040	416	1007	Probe
22.	Normal	97277	87831	79680	Normal
23.	Guess_passwd	53	53	680	R2L
	Total:	**494020**	**145585**	**145572**	

5.6 NB Imbalance Learning within Classes

In line with the findings detailed in Section 4.2.3, NB detection accuracy across minor attacks is recognised as low. Throughout this chapter, the experiments carried out investigate imbalance learning within classes, as opposed to between classes. As detailed in Table 5-2 and Table 5-3, each sub attacks accuracy is provided both prior to and after data resampling when using NB and three different heterogeneous learning algorithms that use NB. It is obvious that the detection accuracy of some sub-minor attacks is behind the poor performance in the models' ability to detect the minor attacks. The overall accuracy concerning the detection of U2R attacks, as seen detailed in Section 4.2.3, are improved owing to the fact that all of U2R sub-attacks' identification accuracy has increased after resampling. However the same cannot be said about R2L sub-attacks when using NB and its variants.

Table 5-2: NB detection accuracy for each sub-minor attack before resample

	NB	Bagging (NB)	Stacking (NB) + RF	Stacking (NB) +Bagging(RF)
Overall Accuracy	78.0%	79.0%	99.77%	99.70%
TP Rate	78.1%	79.0%	99.8%	99.8%
FP Rate	0.1%	0.1%	0.3%	0.3%
Precision	97.2%	97.3%	99.7%	99.7%
back	97.3%	97.2%	97.7%	97.5%
teardrop	99.6%	99.7%	99.6%	99.6%
loadmoudle	55.6%	55.6%	0.0%	0.0%
neptune	99.4%	99.6%	100.0%	100.0%
rootkit	50.0%	50.0%	0.0%	0.0%
phf	**75.0%**	**75.0%**	**25.0%**	**25.0%**
satan	94.2%	94.2%	95.7%	95.4%
buffer_overflow	20.0%	56.7%	50.0%	50.0%
ftp_write	**62.5%**	**62.5%**	**0.0%**	**0.0%**
land	94.7%	94.7%	78.9%	84.2%
spy	**100.0%**	**50.0%**	**0.0%**	**0.0%**
ipsweep	92.8%	95.4%	97.4%	96.8%
multihop	**28.6%**	**14.3%**	**0.0%**	**0.0%**
smurf	99.7%	99.7%	98.4%	98.4%
pod	98.5%	98.5%	97.6%	96.6%
perl	33.3%	33.3%	0.0%	0.0%
warezclient	**48.0%**	**52.9%**	**93.2%**	**92.3%**
nmap	18.4%	18.4%	81.0%	79.1%
imap	**91.7%**	**91.7%**	**75.0%**	**75.0%**
waremaster	**85.0%**	**85.0%**	**75.0%**	**75.0%**
portsweep	72.6%	76.2%	96.4%	95.7%
guess_passwd	**94.3%**	**94.3%**	**94.3%**	**94.3%**
normal	65.0%	79.0%	99.9%	99.9%

Table 5-3: NB detection accuracy of each sub-minor attack after resampling

	NB	Bagging NB	Stacking (NB) + RF	Stacking (NB) +Bagging(RF)
accuracy	79.20%	77.52%	99.89%	99.80%
TP Rate	79.2%	77.5%	99.9%	99.9%
FP Rate	0.2%	0.2%	0.1%	0.1%
Precision	93.6%	93.6%	99.9%	99.9%
back	97.7%	97.8%	99.1%	98.8%
teardrop	99.5%	99.5%	99.8%	99.7%
loadmoudle	77.8%	77.8%	100.0%	100.0%
neptune	99.4%	99.4%	100.0%	100.0%
rootkit	90.0%	90.0%	100.0%	100.0%
phf	**100.0%**	**100.0%**	**100.0%**	**100.0%**
satan	93.9%	93.8%	98.1%	97.4%
buffer_overflow	20.9%	20.9%	100.0%	100.0%
ftp_write	**100.0%**	**100.0%**	**100.0%**	**100.0%**
land	100.0%	100.0%	100.0%	100.0%
spy	**100.0%**	**100.0%**	**100.0%**	**100.0%**
ipsweep	94.3%	94.2%	98.5%	98.4%
multihop	**51.8%**	**52.4%**	**100.0%**	**100.0%**
smurf	99.7%	99.7%	99.8%	96.6%
pod	99.1%	99.1%	100.0%	100.0%
perl	100.0%	100.0%	100.0%	100.0%
warezclient	**53.2%**	**54.7%**	**97.6%**	**96.6%**
nmap	20.0%	20.0%	99.0%	98.3%
imap	**90.8%**	**90.8%**	**100.0%**	**100.0%**
waremaster	**95.4%**	**95.4%**	**100.0%**	**100.0%**
portsweep	74.1%	75.7%	99.7%	99.7%
guess_passwd	**99.4%**	**99.4%**	**100.0%**	**100.0%**
normal	66.0%	62.9%	100.0%	99.9%

5.7 The Factors Underpinning Inadequate Identification of R2L by NB

Across the completion of this experiment, the factors seen to underpin the inadequate identification of R2L attacks through NB has been examined. It is apparent that NB detection for U2R attacks is improved after resampling; however, in the case of R2L, there is no difference. Accordingly, exploration is carried out in relation to the sub-minor attack of KDD cup 99 datasets; this is done in order to establish the factors behind this situation. As has been documented in Table 5-1, eight different sub-minor attacks are incorporated within R2L, namely ftp_write, guess_passwd, imap, multihop, phf, spy, waremaster, and warezclient.

5.7.1 The Accuracy of NB Based Detection Architectures for R2L Sub-Attacks

In the experiments of this section, the emphasis is placed only on the eight sub-minor attacks encompassed within the R2L. The identification accuracy of the NB and its variant on such attacks is explored through eight experiments, as detailed in Table 5-4. Such experiments are centred on using NB, ensemble learning approached (stacking, RF and bagging) and resampling.

As is apparent from the results tabulated in Table 5-4, the identification accuracy of using NB as a stand-alone classifier is significantly improved after resampling across all of the minor attacks, with the exception of the multi-hop and warezclient. This is observed by the fact that the identification accuracy of these two attacks is 51.8% and 53.2% with resampling and 52.4% and 54.7% when using resampling and bagging with NB as the base classifier. This is despite the fact that for the other sub-minor attacks' detection accuracy is over 85% for the same two classification approaches, after using resampling.

A further observation is that without resampling, Stacking (NB) and Bagging (RF) / RF results in poor classification accuracies for all of the sub-minor attacks. The within-class data imbalance is thus having a significant impact on the performance of these classification approaches. It also suggests that there is no point in carrying out Bagging with RF as the base classifier. RF as a standalone learning algorithm performs as an ensemble learner.

Table 5-4: NB detection accuracy for R2L sub-attacks, before and after resampling

	phf	ftp_write	spy	multihop	warezclient	imap	Waremaster	Guess-passwd
NB without resample	75 %	62.5%	100%	28.6%	48.0%	91.7%	85.0%	94.3%
NB with resample	100%	100%	100%	51.8%	53.2%	90.8%	95.4%	99.4%
Bagging NB	75 %	62.5%	50%	14.3%	52.9%	91.7%	85.0%	94.3%
Bagging NB with resample	100%	100%	100%	52.4%	54.7%	90.8%	95.4%	99.4%
Stacking (NB)+ RF	25%	0%	0%	0%	93.2%	75.0%	75.0%	94.3%
Stacking (NB) + RF with resample	100%	100%	100%	100%	97.6%	100%	100%	100%
Stacking (NB) + Bagging(RF)	25%	0%	0 %	0%	92.3%	75.0%	75.0%	94.3%
Stacking (NB) + Bagging(RF) with resample	100%	100%	100%	100%	96.6%	100%	100%	100%

5.7.2 Using Stacking for Improving the Accuracy of Detection of Multihop and Warezclient Attacks

Owing to the poor level of NB based detection accuracy in the case of multihop and warezclient attacks even after bagging and resampling is used, in an effort to improve detection further, an ensemble algorithm —notably stacking—is proposed. The stacking structure that functions effectively involves NB being selected as the base classifier followed by RF as the base classifier of bagging. As shown in Figure 5-3, the detection accuracy of the two sub-minor attacks demonstrates significant improvement through stacking, with the increase seen to be around 50%. The accuracy is 100% for multihop and 96% for warezclient attack detections

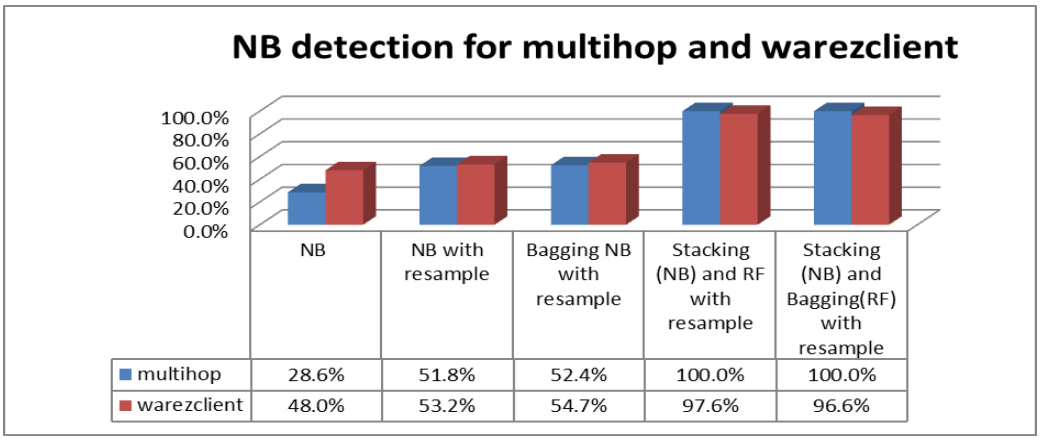


Figure 5-3: NB based detection accuracy for multihop and warezclient sub-minor attacks with resampling.

5.7.3 The Factors Underpinning the Misclassification of Multihop and Wazerclient

5.7.3.1 Multihop Misclassification

Following examination of the NB classifier performance in the identification of multihop attacks, it has been established that, as a result of a number of potential misclassifications with other types of attacks, NB performance in the case of multihop classification does not demonstrate significant increase after resampling. Such misclassifications are detailed in Table 5-5, which provides details of the experiments' misclassification after resampling, with only other sub-minor attacks causing misclassification. It is apparent that, across the various experiments, the multihop is misclassified as Rootkit, Phf, ftp_write, Imap and Warezmaster.

Without the adoption of resampling in the instance of NB, identification accuracy is seen to be 28.6%. Where, 5 samples of 7 multihop attacks are misclassified; notably, 2 were incorrectly classified as Rootkit. In consideration to resampling across NB, a total of 331 out of 639 samples (note: the number of samples has increased from 7 to 639 as a result of resampling) were classified in the correct way, with 308 demonstrating misclassification. Accordingly, in line with this, NB identification accuracy after resampling was found to be 51.8%. Nonetheless, NB based identification as the base classifier of the ensemble learning approach bagging demonstrates some degree of increase of accuracy through the correct classification of 335 out of 639 samples. The most promising outcomes are achieved through stacking, where no misclassification was identified, with the correct classification seen across all 639 samples. Accordingly, when utilised alongside stacking, NB identification accuracy was found to be 100%.

Table 5-5: Multihop - correct classifications

	NB Before Resample	NB after resample	NB bagging	stacking NB + RF	stacking NB + bagging RF
Rootkit	2	-	122	-	-
Phf	-	126	-	-	-
ftp_write	1	91	91	-	-
Multihop	**2/7**	**331/639**	**335/639**	**639/639**	**639/639**
Imap	1	-	-	-	-
Warezmaster	1	91	91	-	-

5.7.3.2 Warezclient Misclassification

In line with the findings garnered through the completion of the experiments, it is apparent that, as a result of various misclassifications, resampling did not achieve significant results in regards the detection of warezclient by NB. Table 5-6 presents an overview of such erroneous classifications, in addition to the sub-minor attacks resulting in such misclassifications. As it is demonstrated in the first experiment, NB prior to the application of resampling, 464 out of 893 samples of warezclient were misclassified. Therefore, warezclient identification accuracy is seen to be 48%. Nonetheless, after resampling, NB detection for the warezclient is enhanced with resampling as 764 out of 1436 warezclient samples are correctly classified meaning an improved detection accuracy of 53.2%. Where the warezclient detection accuracy is slightly increased, 54.7%, with bagging after resample as 785 out of 1436 are correctly detected. Following the application of stacking, where NB provides the foundation and RF is the second classifier, there was a significant improvement in NB performance, with the score achieved 97.6%, with 1401 out of 1436 correctly classified. Nonetheless, in the case of stacking NB and RF rather than stacking NB and bagging, with RF as the base of bagging, NB identification of warezclient was found to be more proficient.

Table 5-6: Warezclient-correct classifications

	NB Before Resample	NB after resample	NB bagging	stacking NB + RF	stacking NB + bagging RF
	Warezclient	**Warezclient**	**Warezclient**	**Warezclient**	**Warezclient**
Loadmodule	25	55	55	-	-
Neptune	-	82	-	-	-
Rootkit	54	-	76	-	-
Phf	-	3	-	-	-
Satan	2	145	3	-	-
Buffer_overflow	137	328	151	-	-
ftp_write	181	-	315	-	-
Spy	-	11	-	-	-
Ipsweep	8	-	8	-	-
Smurf	4	9	-	-	-
Pod	23	-	4	-	-
Warezclient	**429/893**	**764/1436**	**785/1436**	**1401/1436**	**1387/1436**
Nmap	5	4	7	-	-
Warezmaster	19	29	26	-	-
Portsweep	2	2	2	-	-
Normal	4	4	4	35	49

5.8 Summary and Conclusion

NB classifier is the most widely used classifier in state-of-the-art intruder detection systems. The review of literature carried out as a part of this thesis revealed that existing work fails to explain why the NB classifier underperforms in the classification of imbalanced datasets.

The research presented in this chapter has rigorously explored NB classifier behaviour in the presence of imbalanced data, with the conclusion drawn that, for minor attacks, namely R2L and U2R, NB identification requires improvement via the adoption of resampling and ensemble learning methods. Nonetheless, following the adoption of resampling, U2R identification is notably enhanced but a similar improvement is not demonstrated in the detection of R2L attacks.

Following an in-depth examination, this chapter revealed that as a result of various misclassifications with other sub-minor attacks, R2L identification is not enhanced following the application of both bagging and resampling. Accordingly, learning in the presence of imbalanced data within class presents the main focus of this work, as opposed investigating between classes data imbalance. Specifically, the detection accuracy of the R2L sub-minor class was highlighted in this regard. It has been shown that NB classifiers lowest identification accuracies relate to the recognition of multihop and warezclient attacks. These are erroneously

classified as a result of the need to select specific features in the classification. As such, Naïve Bayes should be used in combination with other statistical methods so as to facilitate establishing the most appropriate and valuable aspects for classification. In this regard, stacking is applied in order to bring together NB and bagging.

As the NB classifiers, poor performance in classifying R2L attacks are as a result of erroneous misclassifications between sub-minor attacks, feature selection could be applied in an effort to establish the more individualistic features of the minor attacks and their sub-minor classes. The combined use of pre-processing approaches, feature selection, and resampling should be investigated. In this vein, Chapter 6 provides insight into whether feature selection should be adopted first or whether this should follow resampling. Accordingly, Chapter 7 presents a discussion as to the unique aspects of the attacks, as well as their sub-minor attacks, with experiments carried out in this regard.

Chapter 6 : Impact of the Structure of Data Pre-processing Pipelines on the Performance of Classifiers When Applied to Imbalanced Datasets

6.1 Introduction

Class imbalance is seen to signify a challenge inherent in training datasets that arises due to the lack of proportion of the number of instances present between various classes. Such a challenge can arise in the case of both multi-class and binary-class datasets. In the case of the former, all classes are seen to comprise a particular proportion of samples, with more conventional learning algorithms devised in such a so as to minimise errors across major classes, but with little attention directed towards minority classes, which ultimately results in inadequate accuracy across such classes. In relation to binary-class datasets, it is assumed that the positive samples belong to the minority classes whilst the negative samples belong to the majority classes.

In either of the above cases however, when striving to achieve a greater degree of efficiency or accuracy in prediction of classes, it is common for feature selection to be employed across a dataset. More specifically, the building of feature representations and classification models are enabled through the very likely presence of a limited but nonetheless prominent feature set able to represent instances of a class. Nonetheless, any inadequate feature selection could ultimately result in a lesser degree of discrimination power between classes, meaning that the generated recognition system may have a lower degree of accuracy. Thus, feature selection is recognised as a valuable direction of investigation in the field of machine learning and has accordingly generated significant research interests.

In the application of a supervised classification method, the classifications are dependent on a decision boundary derived from a number of different training samples. Classifier performance quality is usually influenced by the inherent drawbacks of the classification algorithm as well as by the inherent complexity of learning from such samples. The typical reasons for misclassifications to occur are, class overlap, class imbalance, noisy features, low ratios of a sample size to dimensionality, and irrelevant or redundant features been used in the classification. Such hurdles are commonly overcome prior to learning, notably through the application of a pre-processing approach geared towards enhancing the power of the training data. Upon the dual presence of such issues, original training dataset needs to be pre-processed. However, the question remains as to which method should be adopted for pre-processing.

Given the above observations, the research presented within this chapter sets out to investigate the importance of feature selection on the classification accuracy of a classifier, in the presence of class imbalance. The results of this study can provide important recommendations for machine learning and data mining practitioners when designing their classification pipelines, suggesting which combination of pre-processing algorithms would be worthwhile to adopt in applications that need to solve the challenges that arise from imbalance learning problems and more specifically attack classification problem. This chapter centres on the dual utilisation of resampling techniques and feature selection approach within a data preprocessing pipeline of a network IDS, and explores which one, when implemented in what order, would achieve the superior classification results for a given classifier. The two combinations of approaches to be considered will be referred to as RS+FS versus FS+RS; RS+FS may be taken to infer that the training dataset is resampled first, with the features then selected, whereas FS+RS represents the feature selection to proceed to resample.

For clarity of presentation, this chapter is divided into a number of sub-sections. Apart from this section which refers the reader to the research problem that is to be investigated, Section 6.2 presents the experimental methodology and Section 6.3 critically analyse the results obtained. Finally, Section 6.4 summarises the findings and concludes.

6.2 The experimental Setup

The key objectives underpinning the design and implementation of the experiments are the combined application of resampling and feature selection methods, and to determine which order of adoption leads to the most optimal performance of a classifier. Given that there are many different feature selection approaches and many different resampling techniques to be considered, the following two experimental scenarios are adopted in the proposed investigations:

- Scenario 1: All of the feature selection methods are applied in isolation first, using the original dataset. Subsequently, the selected features are appropriately sampled to create the training data set. (FS+RS)
- Scenario 2: A feature selection method is applied to an appropriately sampled dataset. That is, feature selection is carried out on an already sampled dataset. (RS+FS)

The performance of three popular classification algorithms is investigated, with different combinations of feature selection and resampling algorithms, applied under the two scenarios above. The use of NB algorithm is investigated as a Bayesian network, whereas the use of

Random Forest is investigated when used as a decision tree, with stacking as the ensemble learning approach to be adopted. Furthermore, the default values are set as the classifier parameters as implemented by the Weka library for the RF and NB, whilst NB classifier is used in stacking and bagging (RF) implementations.

Irrespective of the fact that improved outcomes could potentially arise as a result of the tuning of the most appropriate and correct parameters, in the case of all experiments, default parameters are adopted across all of the classification algorithms. This, in some way, can lead to baseline performance being maintained as the key base for comparison. In essence, the current work's focus is centred not only on the analysis of the benefits and disadvantages associated with the use of the categorisation algorithm but also the combined effect associated with the resampling and feature selection in relation to imbalance learning. As such, in the case of all of the experiments, for all of the classification algorithms, the default parameters are adopted, as well as for feature selection. With a direct link to the aforementioned, so as to determine the most prevalent and noteworthy differences with regards to the findings to be concluded through the methods to be implemented; there is a need for statistical analysis to be carried out. Importantly, there is also a need for an experimental stage to be included, which is both detailed and wide-ranging, making use of a 10-fold stratified cross-validation test to analyse the performance of a number of different classification algorithms, 7 feature selection methods (6 of which are acknowledged as being ranking approaches, whereas one is seen to be the best first method—cfs), and the random under-resampling approach (widely recognised as achieving a high level of balance across specific data sets, notably via the random negation of instances from the majority class) as the only sampling method. Eventually, a number of different approaches are taken into account, resulting in the completion of 42 experiments in total.

6.3 Experimental Result and Analysis

6.3.1 The Selected Features

The features selected under the two experimental scenarios, when different feature selection algorithms are used, are presented in this section. As detailed in Table 6-1 and 6-2, the feature ranking of each algorithm differs from one algorithm to the next (Note: the selected features are presented in rank order in both tables, with the first listed feature having the highest rank). Furthermore, when examining each algorithm, there is a different selection of features and

rankings when the two scenarios are used, i.e. depending on whether the feature selection method is implemented on the resampled dataset or on the original dataset before resampling.

It is noted that all of the different feature selection algorithms decide on a different number of selected features. This number is based on the parameter selections associated with each feature selection algorithm. In our experiments, we used default parameters for all feature selectors. However, it is possible to keep the subsequent computational cost of using the selected features in classification fixed by selecting the best-N features for all classifiers. For example when N=5, CFs selects features f1, f3, f4, f5, and f6 when using scenario-1 but uses features f3, f4, f5, f6, and f7. When using scenario-2. Feature f1 has been replaced by f7 but at a different rank order. In other words, the highest-ranked feature under scenario-1 is no longer a discriminant feature under scenario-2. This is an interesting and powerful observation. Further, the best five features that are selected by the 'correlation' feature selection approach are f29, f33, f34, f38, f39 when using scenario-1 and f29, f34, f38, f25, f39 when using scenario-2. The comparison of CFs and 'correlation' feature selection approaches reveal that they have resulted in a completely different set of features. This indicates that it will also be important to determine the accuracy of classification that will result from these selections, prior to one deciding on the final set of features.

Which selection of features should be selected can only be concluded based also on the classification accuracy a particular feature selection achieves. Therefore section 6.3.2 carries out experiments to determine classification accuracies obtainable when using different classifiers, when a particular feature selection is used, before or after resampling.

Table 6-1: The selected Feature in case of RS+FS

Resampling +Feature Selection		
NO.	**Feature Selection Method**	**Feature Ranking**
1.	**CFs**	f1,f3,f4,f5,f6,f7,f8,f10,f12,f23,f25,f26,f29,f30,f32,f33,f34,f35,f36,f37, f38,f39
2.	**Correlation**	f29,f33,f34,f38,f39,f25,f26,f4,f23, f12,f32,f31
3.	**GainRatio**	f7,f8,f11,f14,f18,f15,f13,f10,f17,f4,f26,f25,f9,f19,f39,f16,f12,f38,f30
4.	**InfoGain**	f5,f3,f6,f23,f33,f35,f30,f29,f4,f34, f38,f39
5.	**OneR**	f5,f3,f30,f29,f23,f6,f4,f35,f34,f33, f38,f39,f25,f26,f12
6.	**ReflieF**	f3,f4,f38,f12,f26,f25,f39,f34,f33,f29,f32,f36,f2,f23
7.	**SymmetricalUncert**	f3,f4,f30,f38,f29,f39,f5,f25,f26,f12,f35,f6,f23,ff34,f33,f36,f37,f32,f1

Table 6-2: The selected feature in case of FS+RS

Feature Selection + Resampling		
NO.	**Feature Selection Method**	**Feature Ranking**
1.	**CFs**	f3,f4,f5,f6,f7,f8,f10,f12,f23,f25,f26,f29,f30,f32,f33,f34,f35,f36,f37,f38,f39
2.	**Correlation**	f29,f34,f38,f25,f39,f26,f23,f4,f33, f12,f3,f32
3.	**GainRatio**	f8,f7,f13,f11,f26,f25,f4,f10,f30,f12,f39,f38
4.	**InfoGain**	f5,f3,f30,f29,f23,f4,f35,f34,f33,f6,f38,f25,f39,f26,f12
5.	**OneR**	f30,f29,f5,f3,f23,f35,f4,f34,f33,f38,f25,f39,f26,f6,f12,f36,f32
6.	**ReflieF**	f4,f3,338,f12,f26,f25,f39,f33,f34,f29
7.	**SymmetricalUncert**	f30,f4,f25,f29,f38,f3,f39f12,f5,f35,f34,f6,f23,f33

6.3.2 Impact of Resampling and Feature Selection on Classification Accuracy

In this section, we investigate the performance of classifiers, NB, Random Forest and Stacking under the two scenarios mentioned in Section 6.2. Upon examination, it would become apparent which order of application could achieve the most promising results (scenario-1 or 2) and through the adoption of which feature selection method/s.

With the application of NB in the role of learner, it is seen through the results in Table 6-3 that, in examining incorrectly classified instances (ICI), Fs+Rs is seen to perform better than Rs+FS

in the majority of the cases; this is seen to be the case with the exception of the application of the feature selection methods of GainRatio, oneR, and ReliefF. In the instance of using Random Forest classifier, on the other hand, in the majority of instances, the most optimal results are garnered by Rs+FS as opposed to FS+RS. Such a finding is rationalised with consideration of the inherent capacity of a decision tree to choose the most appropriate features throughout the training. Stacking performance, in the majority of instances, is seen to achieve greater outcomes in the scenario of RS +FS than in Fs+Rs.

When considering feature selection from the algorithm perspective, the majority are seen to result in the most optimal classification in the sequence in which resampling is adopted prior to feature reduction, Rs+Fs. Nonetheless, the most optimal performance of Naives byase is recognised as garnered through the application of the symmetricalUncert method, where feature reduction is carried out prior to resampling, achieving an overall accuracy of 80.6%, with 19.4% ICI. Moreover, in specific regards RF, the most optimal performance is seen to be achieved through the application of SymmetricalUncert, achieving 99.98% accuracy with only 36 points of misclassification across the Rs+FS order.

The final row provides insight into the degree to which average features are mainly ranked in line with the running feature selection approaches influencing the used classifier performance. The most optimal performance is when resampling is adopted after feature selection (FS+RS) and is achieved through NB and RF, whereas the most optimal identification is achieved by Stacking in regards Rs+Fs. Owing to the fact that stacking in RS+FS case, to some degree, outperforms Fs+Rs, with only 18 more instances being correctly classified. Whilst, 388 were incorrect through the adoption of Fs+Rs; this was seen to be 370 instances under consideration of Rs+Fs.

Table 6-3: Classifiers' Performance on the Application of Resampling and Feature Selection Approaches

		NaivesBayes		RandomForest		Stacking	
		Acc	**ICI**	**Acc**	**ICI**	**Acc**	**ICI**
Cfs	**Fs+Rs**	72.4% (105324)	**27.6% (40248)**	99.8% (145258)	.2% (314)	98.3% (143074)	1.7% (2498)
	Rs+fs	69.9% (101893)	30% (43679)	99.98 (145535)	**(.03) 37**	99.7% (145109)	**.3% (463)**
Correlation	**FS+Rs**	58.1% (84507)	**41.9 (61065)**	98.9% (144033)	1% (1539)	98.5% (143346)	1.5% (2226)
	Rs+fs	57.7% (83992)	42.3% (61580)	99% (144175)	**.96% (1397)**	98.6% (143587)	**1.4% (1985)**
GainRatio	**Fs+Rs**	34% (49526)	65.98 (96046)	95.6% (139188)	4.4% (6384)	95.1% (138400)	4.9% (7172)
	Rs+Fs	36% (52369)	**64% (93176)**	95.8% (139441)	**4.2% (6131)**	95.6% (139136)	**4.4% (6436)**
Infogain	**Fs+Rs**	77.3% (112605)	**22.6% (32967)**	99.9% (145431)	**.1% (141)**	99.4% (144703)	**.5% (869)**
	Rs+Fs	71.4% (103917)	28.6% (41655)	99.8% (145311)	.18% (261)	98.97% (144069)	1% (1503)
OneR	**Fs+Rs**	65.9% (95914)	34.1% (49658)	99.1% (144314)	.9% (1258)	98.98% (144090)	1% (1482)
	Rs+Fs	77.4% (112605)	**22.6% (32967)**	99.9% (145431)	**.1% (141)**	99.4% (144703)	**.5% (869)**
ReliefF	**Fs+Rs**	65.9% (95914)	34.1% (49658)	99.1% (144314)	**.9% (1258)**	98.7% (143669)	**1.3% (1903)**
	Rs+Fs	66% (96026)	**34% (49546)**	99% (144166)	.97% (1406)	98.6% (143486(	1.4% (2086)
SymmetricalUncert	**Fs+Rs**	80.6% (117333)	**19.4% (28239)**	99.98% (145535)	.03 (37)	99.5% (144865)	.5% (707)
	Rs+fs	78.5% (114289)	21.5% (31283)	99.98% (145536)	**.02% (36)**	99.6% (145020)	**.4% (552)**
Average	**Fs+Rs**	82.2% (120563)	**17.2% (25009)**	99.98 (145536)	**.02% (36)**	99.7% (145184)	.3 (388)
	Rs+Fs	80.34% (1169530	19.7% (28169)	99.98% (145537)	.02% (35)	99.7% (145202)	**.3% (370)**

6.4 Summary and Conclusion

When examining the KDD cup' 99 dataset, there is a lack of balance across class distribution, with a requirement to decrease feature space dimensionality. The current study explored whether or not feature selection in line with a particular approach could overcome dataset skewness or whether, in fact, the contrasting pipeline would demonstrate greater performance. The results that underwent statistical analysis revealed that both pipelines are recognised as worthy of consideration, when there is a need of it, to get an optimal classification model. This is concerned with the classifier adopted and the feature selection methods applied. In a greater

number of cases, feature selection after resampling approach demonstrates greater performance in comparison to the opposing pipeline.

When considering feature reduction, the conclusion drawn is that in line with the methods of feature selection applied, in addition to whether or not the dataset is resampled, the features ultimately chosen differ from one method to the next. In the majority of instances, there is greater performance demonstrated by the random forest algorithm when there is the adoption of features reduction following resampling. This is explained in consideration to the random forest being recognised as comprising various decision trees. All decision tree nodes present a condition on an individual feature, which has been devised in such a way so as to divide the dataset in half to achieve comparable response values across the set. Owing to the fact that the dimensionality of the feature space is reduced as a result of feature selection, it is that the random forest performance could be hindered. Owing to the fact that, following feature reduction, the random forest will build the decision trees across a small number of imbalance dataset features, where the samples across some of the categories are notably lacking. The random forest needs to ensure there is an adequate number of samples when training so as to ensure the amount by which each feature decreases the tree's weighted impurity can be calculated.

Otherwise stated, there are a number of different aspects to consider when applying a ranking focused on impurity: primarily, when applying feature selection that is essentially focused on reducing impurity, there is the presence of bias in relation to variables with additional categories. Secondarily, when the dataset has two (or more) correlated features, from the framework's perspective, any such correlated features are applied as a predictor, without any degree of preference for one over another. However, upon the adoption of one, there is a notable decrease in the importance of others owing to the fact that the impurity removed has previously been removed through the preliminary feature, which then leads to lower reported importance. When seeking to apply the feature selection approach in order to decrease overfitting, this does not pose a problem owing to the fact that the removal of features that are, in the main, duplication by other features is a rational and logical step. However, when it comes to data interpretation, on the other hand, inaccurate conclusions are drawn; suggesting that one of the variables is positioned as a strong predictor whereas the other in the group lacks value and importance. In actuality, it could be that they are similar in regards to their link to the response

variable. This same observation is witnessed in regards the stacking approach when considering that stacking is essentially centred on the random forest as the underpinning of bagging.

In the case of Naïve Bayes, put simply, a specific feature of a class being absent or presence is not related to the absence or presence of another feature. Where each attribute's probabilities notably computed and evaluated on an individual basis away from the training dataset. Naïve Bayes' performance is seen to decline when the features within the data are significantly correlated. This is owing to the fact that such highly correlated features receive two votes in the model, which therefore means their value is seen to be exaggerated. Accordingly, it is necessary that the link between pairwise attributes is assessed through the application of a correlation matrix, ensuring that any features that are highly correlated are removed. Owing to the presence of correlated features in the KDD cup dataset, Naïve Bayes performance across FS+RS is seen to be better in comparison to the contrasting pipeline. Importantly, redundant features are first reduced then the dataset distribution is balanced. As by resampling the Naive Bayes achieves a suitable volume of data in order to facilitate the development of insight into the probabilistic relationship across features in isolation with the output variable.

Following an examination into the appropriate sequence of methods in this chapter, i.e. scenario-1 vs scenario-2, Chapter 7 will focus on the particular aspect of the accuracy of detection of each attack and their sub-minors. Chapter 7 will, therefore, implement resampling first, followed by feature.

Chapter 7 : Feature Selection for the Minor Attacks and Its Sub-minor Attacks

7.1 Introduction

IDSs are recognised as concerned with a large volume of data that comprises different network traffic trends, where dataset patterns could be acknowledged by a number of different feature sets, i.e. attributes that are characterised as one factor fundamental to multi-dimensional feature space. A trend is seen to involve a number of different factors that are irrelevant, and which are therefore recognised as causing the efficiency of training and testing operations to decline, with classification sometimes impacted as a result of the presence of higher mathematical complexity. When adopting a practical standpoint, on the other hand, it might be advantageous for various features to be kept to a minimum; this could help to ensure computational costs and the building complex of the classifier is decreased. As such, the performance of the systems discussed thus far could be improved through the addition of a number of different phases, notably alongside dimensionality reduction as one key element inherent in the phase of pre-processing; this is applied in an effort to remove from the dataset any insignificant aspects. When considering the reduction of dimensionality, as shown through feature extraction and feature selection, success has been achieved by the following implementation in learning with data-mining and machine-learning in order to solve the issue. Moreover, Feature Extraction (FE) methods are focused on transferring input features, which are then incorporated within a new feature set; on the other hand, in the case of the original input data, the most valuable aspects are established via FS algorithm applied. This study directs its attention towards feature selection.

Upon examining the KDD Cup 99 dataset, with consideration also centred on feature categorisation, it is seen that there are a total of four different groups of features as listed in Table 7-1; the first group is seen to include those features/aspects referred to as labels 1–9, which signify the pivotal underlying elements of different TCP connections; the second group, which notably spans features 10–22, is seen to be linked with content features, whilst the third group, ranging features 23–31, includes traffic features, which are calculated with the adoption of a two-second timeframe; finally, those labelled 32-41 make up the fourth group and include traffic features determined through the adoption of the two-second time window spanning destination to host.

Table 7-1: KDD cup 99 dataset features category

Category	Features Name	Description
C1	1. **Duration**	Length (number of seconds) of the connection
	2. **Protocol-type**	Type of the protocol, e.g. tcp, udp, etc.
	3. **Service**	Network service on the destination, e.g., http, telnet, etc.
	4. **Flag**	Normal or error status of the connection
	5. **Src-bytes**	Number of data bytes from source to destination Number of data bytes from destination to source
	6. **Dst-bytes**	
	7. **Land**	1 if connection is from/to the same host/port; 0 oth- erwise
	8. **wrong-fragment**	Number of "wrong" fragments
	9. **Urgent**	Number of urgent packets
C2	10. **Hot**	Number of "hot" indicators
	11. **Num-failed- logins**	Number of failed login attempts
	12. **Logged-in**	1 if successfully logged in; 0 otherwise
	13. **Num- compromised**	Number of "compromised" conditions
	14. **Root-shell**	1 if root shell is obtained; 0 otherwise
	15. **Su-attempted**	1 if "su root" command attempted; 0 otherwise
	16. **Num-root**	Number of "root" accesses
	17. **Num-file- creations**	Number of file creation operations
	18. **Num-shells**	Number of shell prompts
	19. **Num-access- files**	Number of operations on access control files
	20. **Num- outbound-cmds**	Number of outbound commands in an ftp session
	21. **Is-hot-login**	1 if the login belongs to the "hot" list; 0 otherwise
	22. **Is-guest-login**	1 if the login is a "guest" login; 0 otherwise
C3	23. **Count**	Number of connections to the same host as the current connection in the past 2 s
	24. **Srv-count**	Number of connections to the same service as the current connection in the past 2s
	25. **Serror-rate**	% of connections that have"SYN"errors(same-host connections)
	26. **Srv-serror-rate**	%of connections that have "SYN"errors(same-service connections)
	27. **Rerror-rate**	% of connections that have "REJ" errors(same-host connections)
	28. **Srv-rerror-rate**	%of connections that have "REJ" errors (same-service connections)
	29. **Same-srv-rate**	% of connections to the same service (same-host connections)
	30. **Diff-srv-rate**	% of connections to different services (same-host connections)
	31. **Srv-diff-host-rate**	% of connections to different hosts (same-service connections)
C4	32. **Dst-host-count**	Count for destination host
	33. **Dst-host-srv-count**	Srv-count for destination host
	34. **Dst-host-same-srv-rate**	Same-srv-rate for destination host
	35. **Dst-host-diff-srv-rate**	Diff-srv-rate for destination host
	36. **Dst-host-same-src-port-rate**	Same-src-port-rate for destination host
	37. **Dst-host-srv-diff-host-rate**	Diff-host-rate for destination host
	38. **Dst-host-serror-rate**	Serror-rate for destination host
	39. **Dst-host-srv-serror-rate**	Srv-serror-rate for destination host
	40. **Dst-host-rerror-rate**	Rerror-rate for destination host
	41. **Dst-host-srv-rerror-rate**	Srv-serror-rate for destination host

Owing to the apparent lack of balance in the Kdd cup'99 dataset, it is required that the accuracy of the classification and various characteristics of the minority classes be examined, which is viewed as important. Not dissimilar to other imbalanced datasets, there is a high degree of

overall classier accuracy, whereas, on the other hand, in the case of the minor classes, along with its corresponding sub-minor classes, there is a low level of true positive rate.

As has been highlighted in the previous chapter, one of the key factors underpinning the inadequate identification of the R2L minor attack is sub-minor attacks' misclassification. Accordingly, throughout this chapter, an analysis will be carried out with regards to the minor attacks' relevance of features and the corresponding sub-attacks. During the completion of the experiments, the relevance between binary class and multi-class issues and their features will provide a point of focus.

For clarity of presentation, this chapter is divided into four subsections. Apart from this section that introduced the research problem addressed in this chapter, Section 7.2 presents details of the experimental setup used for the investigation. Section 7.3 presents the results and a detailed analysis of the results. Finally, section 7.4 summarises the outcomes of the investigation and concludes the findings.

7.2 Experimental Setup

When seeking to establish the overall level of relevance of the sub-minor attack (U2R and R2L) features on the classification accuracy and to further identify the extent to which the classifier accuracy will depend on the selected features when the dataset is binary-class or a multi-class dataset it is proposed to adopt two different experimental approaches as below for the two types of minor attacks:

A. Establish the Attack Features of U2R

1. The dataset is a binary class dataset, containing only U2R attacks and normal data.
2. The dataset is multi-class, encompasses all U2R sub-minor attacks, as well as normal data.
3. The dataset is a binary class dataset, containing only Buffer Overflow attack and normal data.
4. The dataset is a binary class dataset, containing only loadmodule attacks and normal data.
5. The dataset is a binary class dataset, containing only Perl attacks and normal data.
6. The dataset is a binary class dataset, containing only Rootkit attacks and normal data.

B. Establish the Attack Features of R2L

1. The dataset is a binary class dataset, containing only R2L attacks and normal data.
2. The dataset encompasses all R2L attacks, sub-minor attacks, as well as normal data.
3. The dataset is a binary class dataset, containing only FTP attacks and normal data.
4. The dataset is a binary class dataset, containing only Password Guessing Attacks and normal data.
5. The dataset is a binary class dataset, containing only IMAP attack and normal data.
6. The dataset is a binary class dataset, containing only Warezmaster attack and normal data.
7. The dataset is a binary class dataset, containing only Warezclient attack and normal data.
8. The dataset is a binary class dataset, containing only Spy attacks and normal data.
9. The dataset is a binary class dataset, containing only Multi-Hop attacks and normal data.
10. The dataset is a binary class dataset, containing only PHF attacks and normal data.

Importantly, those feature selection algorithms applied across the scenarios listed above are the various feature selection approaches investigated in the previous chapter without reference to their ability to classify attacks in imbalanced datasets. In this chapter, the use of the selected by using the various feature selection algorithms in classifying attacks using the NB classifier will be investigated in detail. It was decided to use NB as the choice of the classifier as this is the most frequently used classifier by the research community involved in research into network intruder/attack detection.

7.3 Experimental Results and Analysis

7.3.1 Characteristics of U2R Attacks

7.3.1.1 U2R Attack vs. Normal Data Experiment

Table 7-2 provides an overview of the U2R attack feature ranking, through which it is seen that the most critical aspect in identifying U2R attacks requires consideration of the aspects relating to content characteristics. Importantly, the most commonly featured characteristic through the applied algorithm is recognised in the content feature group—notably f10, f13, f14, f16, f17 and f18; referred to as Hot, Num-compromised, Root-shell, Num-root, Num-file-creations, and Num-shells, respectively. As can be seen when reviewing the results, individual TCP connection characteristics adopt a key part in identifying U2R attacks, as in the cases of f1, f3,

f5, and f6, referred to as Duration, Service, Src-bytes and dst-bytes, respectively, where all of these are commonly ranked in line with the approaches relating to the feature selection approach used. As a summary, when seeking to identify U2R attacks, there is no need to take into account the aspects of the traffic, calculated through the application of a two-second time window. Owing to the fact that the ranked features are not viewed as belonging to its category, nonetheless, traffic features calculated through the application of a two-second time window, notably travelling from destination to host, need to be taken into account.

Table 7-2: Ranked features of U2R attack

Features of U2R attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	Cfs (BestFirst)	fF10, f11,f13,f14,f17,f27,f33,f38	90.1%(79218)
2.	Correlation	f14,f33,f17,f36,f10,f18,f3	94.7%(83196)
3.	GainRatio	f14,f13,f17,f10,f9,f18,f11,f16	**97.4%(85574)**
4.	InfoGain	f6,f3,f5,f33,f1,f10,f14,f32	93.5%(81754)
5.	OneRAttribute	f6,f1,f3,f5,f14,f6,f7,f8	95.3%(83775)
6.	ReliefFAttribut	f3,f33,f34,f36,f31,f32	90.4%(79443)
7.	SymmetricalUncert	f14,f13,f10,f17,f1,f33,f3,f16	96.5(84834)
8.	Average	f1,f3,f5,f6,f10,f13,f14,f16,f17,f18,f32,f33,f36	93.4(82368)
9	All features	f1-f41	92.7%(81447)

7.3.1.2 U2R Sub-Minor Attacks vs. Normal Data Experiment

Through this particular investigation, there will be a summary pertaining to the characteristics recognised as needing to be taken into account when identifying sub-minor attacks (see Table 7-3). It suggested that the outcomes arrived through the application of the multi-class U2R dataset differs to that of the binary-class U2R dataset. This is predominantly as a result of various new aspects being ranked frequently, whereas some features have been completely ignored. Across U2R sub-minor attacks, it is maintained that the destination–host traffic characteristics category is insignificant in the identification of U2R sub-minor attacks. The only characteristic ranked through the applied technique is that of a dst_host_srv_count feature, f33, whereas various other characteristics need to be taken into account in comparison to the binary-class case, such as Num-failed-logins, f11, which is an ongoing aspect viewed as significant when establishing the sub-minor occurrence of U2R attacks.

Table 7-3: Ranked features of U2R sub-minor attacks

Features of U2R Sub-minor attacks			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	Cfs (BestFirst)	f1,F10,f11,f14,f16,f17,f18,f29,f40	15.1%(12387)
2.	Correlation	f14,f18,f17,f33,f3	91.5%(75268)
3.	GainRatio	f18,f14,f17,f9,f13,f16,f11,f10	**94.85(78012**)
4.	InfoGain	f6,f5,f1,f3,f33,f17,f14	90.8%(74705)
5.	OneRAttribute	f6,f1,f5,f17,f16,f18,f14,f13,f10	14.9%(12261)
6.	ReliefFAttribut	f3,f33,f34,f32,f36,f14	89.3%(73534)
7.	SymmetricalUncert	f17,f14,f18,f16,f10,f13,f1	94.7%(77865)
8.	Average	f1,f3,f5,f6,f10,,f11,f13,f14,f16,f17,f18,f33	91.8%(75480)
9.	All features	f1-f41	**95.5%(78579**)

7.3.1.3 Buffer Overflow Attack vs. Normal Data Experiment

The results of this experiment are tabulated in Table 7-4. It shows that with regards to the buffer overflow attack, the categorisation of TCP connection feature plays a fundamental part in the identification. Owing to the fact that the majority of the common features ranked through the applied methods belong to this particular category—which is notably f1, f3, f5 and f6, referred to as duration, Services, Src-bytes and Dst-bytes, respectively—the unique characteristic ranked by at least one approach and is recognisable only in the case of this attack is that of Serror_rate, f25, which is seen to belong to the category feature of Traffic. This confirms that through establishing the overall percentage of links with some degree of SYN error identifiable in the same-host connection, identifying the frequency and presence of buffer overflow is possible. It can be seen that the characteristics hot (f10) and Iroot_shell (f14) are the most commonly identified aspect, which is a view supported through the application of six of the adopted approaches.

Table 7-4: Ranked features of Buffer Overflow attack

Features of Buffer overflow attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	Cfs (BestFirst)	f10,f13,f14,f17,f32,f33	99.7%(80131)
2.	Correlation	f14,f17,f10,f33,f36,f25,f32,f3	**99.8%(80183)**
3.	GainRatio	f14,f13,f17,f10,f25	99.2%(79690)
4.	InfoGain	f6,f5,,f3,f10,f1,f14,f13,f33	95.6%(76776)
5.	OneRAttribute	f6,f5,f14,f1,f13,f10,f3,f17	97.8%(78579)
6.	ReliefFAttribut	f3,f33,f32,f36,f12	99.8%(80156)
7.	SymmetricalUncert	f14,f13,f10,f17,f1	99.3%(79763)
8.	Average	f1,f3,f5,f6,f10,f13,f14,f17,f25,f32,f33,f36	99.5%(79967)
9.	All features	f1-F41	99.5%(79946)

7.3.1.4 Loadmodule Attack vs. Normal Data Experiment

As detailed in the table 7-5, when seeking to identify attacks of the type Loadnodule, the most basic aspect of the individual content group needs to be determined, with the majority of the most commonly ranked aspects within this group recognised as f10, f13, f14, f16, f17 and f18, namely Hot, Num-compromised, Root-shell, Num-root, Num-file-creations and Num-shells, respectively. One of the continuous aspects, referred to as Dst-host-srv-diff-host-rate, f37, is recognised as a characteristic individual to the identification of the loadmodule attack. This particular aspect has been graded by at least two of the different approaches applied and is concerned with the different host rate for the host destination. Nonetheless, as has been demonstrated by six of the methods, the most commonly identified characteristic is that of lnum_file_creations (f17).

Table 7-5: Ranked features of Loadmodule Attack

Features of Loadmodule Attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	Cfs (BestFirst)	f14,f17,f32,f33	99.6% (80015)
2.	Correlation	f14,f18,f17,f37,f36,f33	95.1%/
3.	GainRatio	f18,f14,f13,f17,f10,f16	99.3%(79772
4.	InfoGain	fF6,f3,f33,f5,f1,f32,f36,f37,f17,f10	99.4%(79870)
5.	OneRAttribute	f6,f1,f17,f14,f18,f37,f3,f13,f10,f5	95.55(76699)
6.	ReliefFAttribut	f3,f33,f32,f35,f34,f36	**99.7%(80093)**
7.	SymmetricalUncert	f14,f17,f18,f10,f13,f33,f16,f32	99.7%(80067)
8.	Average	f1,f3,f5,f6,f10,f13,f14,f16,f17,f18,f32,f33,f36,f37	99.5%(79907)
9.	All Features	f1-f41	99.3%(79762)

7.3.1.5 Perl Attack vs. Normal Data Experiment

As shown when reviewing the results of Table 7-6, it is apparent that, in the identification of the Perl attack, it would not be relevant to take into account the category of traffic features (f23-f31) owing to the observation that none of the approaches are seen to rank any features within this particular group. When examining the most valuable of characteristics—as determined through the various techniques applied—which, when positioned in the first ranking warrants consideration in relation to connection length (Duration), number of data bytes from destination to source (Dst-bytes) and whether or not the root-shell is obtained (Root-shell). The ranking of all of these features is carried out in the first instance and on more than one occasion. Owing to the fact that Duration, f1, is ranked through Correlation and SymmetricalUncert methods throughout the first instance of their rank, f6, Dst-bytes, was found to secure the highest rank through the application of both InfoGain and OneRAttribute

algorithms. Importantly, nonetheless, the most dominant feature amongst the ranking features is determined as being Correlation and GainRatio rank, the Root-shell, f14. However, when examining the U2R attacks and their sub-minor categories, the Perl attack is viewed as being the only one not ranking the hot feature, f10, as important in line with identification. With Perl attacks unable to be identified through establishing the instances of directory access, It is imperative that that lroot_shell (f14) and lnum_shells (f18) features are taken into consideration as these are considered valuable across all methods applied.

Table 7-6: Ranked features of Perl attack

Features of Perl attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	Cfs (BestFirst)	f1,f14,f16,f17,f18	100%
2.	Correlation	f14,f18,f17,f34,f33,f3	99.99 (80313)
3.	GainRatio	f14,f18,f17,f16,f1,f33,f34	99.99%(80312)
4.	InfoGain	f6,f16,f14,f17,f18,f1,f3,f5,f34,f33	99.99%(80314)
5.	OneRAttribute	f6,f16,f14,f17,f18,f1,f3,f5,f40	99.99%(80313)
6.	ReliefFAttribut	f18,f14,f3,f34,f33,f12	99.98%(80297)
7.	SymmetricalUncert	f1,f14,f16,f17,f18	100%
8.	Average	f1,f3,f5,f6,f14,f16,f17,f18,f33,34	99.99%(80314)
9.	All Features	f1-f41	99.99%(80304)

7.3.1.6 Rootkit Attack vs. Normal Data Experiment

As shown in Table 7-7, when seeking to establish the rootkit attack, the calculated traffic characteristics through the utilisation of a two-second time window is not viewed as relevant for consideration, with none of the algorithms providing it a ranking. Nonetheless, there is a need for there to be the determination of rootkit occurrence, which is achievable through the characters within the TCP connection and content group. Such aspects are seen to span f1–f9 in the group of TCP connection, whilst range f10–f22 is in relation to content. As can be seen, the individual features ranked for rootkit attack—notably by at least four of the commonly implemented methods—are an urgent feature, F9 which makes up the urgent packets, and Num-failed-logins feature, f9, which is seen to represent the number of failed login attempts. Nonetheless, six of the methods applied showed agreement in regards to the importance of the rootkit attack detection, lnum_root (f16).

Table 7-7: Ranked features of Rootkit attack

Features of Rootkit attack			
No	**Feature Algorithms**	**Features Rank**	**NB**
1.	Cfs (BestFirst)	fF11,f13,f16,f17,f33	**99.5%(79917)**
2.	Correlation	fF14,f9,f11,f33,f34,f16,f17,f39,f3	93.1%(74744)
3.	GainRatio	f9,f14,f11,f13,f16,f17,f10	99.2%(79679)
4.	InfoGain	f5,f6,f3,f1,f33,f16,f35,f34	96.6%(77571)
5.	OneRAttribute	f5,f6,f1,f16,f13,f14,f3,f9,f11	93.3%(74958)
6.	ReliefFAttribut	f3,f33,f34,f36,f31,f32	92.1%(73959)
7.	SymmetricalUncert	f13,f16,f14,f17,f9,f11,f10,f33,f1	99.5%(79910)
8.	Average	f1,f3,f5,f6,f9,f10,f11,f13,f14,f16,f17,f33,f34	94.2%(75637)
9.	All Features	f1-f41	96.4%(77417)

7.3.2 Characteristics of R2L Attacks

7.3.2.1 R2L Attack Vs. Normal Data Experiment

In line with the results obtained (see Table 7-8), it is possible to identify an R2L attack by examining those characteristics that are categorised into the content, TCP connection, and traffic feature (destination to host) group. Through the support of the applied methods in with regards to these features, which are acknowledged as being significant, the ranked features across the TCP connection include Duration, Service, Src-bytes, Dst-bytes and urgent, which are represented through f1, f3, f5, f6, and f9, respectively.

Importantly, however, in the content group, the ranked characteristics include f10, f1 and f22, recognised as Hot, Num-failed-logins and is- guest-login, respectively. In consideration to the ranked aspects of the traffic features group, this takes into account the destination–host two-second time window, this notably progresses F33, f36 and f37, recognised as Dst-host-srv-count, Dst-host-same-src-port-rate, and Dst-host-srv-diff-host-rate, respectively. Nonetheless, it is observed via the experimental results that the various features across the traffic feature group, with specific association to services and host, were not seen to play a valuable part in the identification of R2L, with the exception of the ongoing characteristics that measure and calculate the instances of connection to the same service, with the present connection in the past 2s labelled as valuable through the approach of SymmetricalUncert. This particular aspect is referred to as Srv- count, f24. Nonetheless, the most commonly ranked characteristic, as determined through all of the techniques applied, is that of service (f3) with hot (f10) following as a result of six of the methods applied.

Table 7-8: Ranked features of R2L attack

Features of R2L attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f3,f9,f10,f22	98.8%(87795)
2.	Correlation	f10,f22,f33,f36,f3,f37,f5	98.85(87749)
3.	GainRatio	f22,f10,f11,f9,f13,f3	**98.9%(87842)**
4.	InfoGain	f5,f3,f6,f33,f10,f36,f24,f37	98.65(87579)
5.	OneRAttribute	f5,f10,f6,f3,f22,f1,f11,f39,f4	98.7%(87695)
6.	ReliefFAttribut	f14,f3,f32,f33,f2,f36	96.1%(85357)
7.	SymmetricalUncert	f10,f22,f3,f33,f5,f6,f1,f24,f36	98.7%(87706)
8.	Average	f1,f3,f5,f6,f9,f10,f11,f22,24,f33,f36,f37	97.8%(86920)
9.	All Features	f1-F41	98.4%(87439)

7.3.2.2 R2L Sub-Minor Attacks vs. Normal Data Experiment

As illustrated in Table 7-9, hot feature, referred to as f10, plays a prominent role in the case of the ranked lists of six of the seven individual feature selection approaches adopted, where such a characteristic is seen to manage the number of urgent packets. The only method to fail to rank it highly is that of ReliefFAttribute. As has been clearly demonstrated, those aspects that are seen to belong to the traffic feature group (f23–f31), and which show support for the services and host connection, are not viewed as being significant in line with the identification of R2l sub-minor attacks. In terms of the ranking given by those methods adopted, the most common features are identified as included in the content, traffic feature, and TCP connection groups.

In the features of the content group, these are established as f1,f3,f5,f6, whereas those of the feature content group these are identified as f10, f11, f14 and f19, whilst the traffic feature group incorporates f33, f36, f38 and f39, which are calculated with the application of the destination–host two-second time window. Converse to those aspects supported in relation to the identification of R2L, the identification of the sub-minor attacks requires various individual features. Such features, which are notably not highlighted in R2L identification, include f14, f19, f38 and f39, which are referred to as Root-shell, Num-access-files, Dst-host-serror-rate, and Dst-host-srv-serror-rate. A number of the aspects which have not been ranked in terms of R2L identification of sub-minor attacks but are ranked for the R2l attack include Urgent, Is-guest-login and Dst-host-srv-diff-host-rate, which are otherwise referred to as f9, f22, and f37, respectively. In addition, as shown through the prior case, f10 (hot) is acknowledged as being the most commonly ranked feature by the various feature selection approaches applied.

Table 7-9: Ranked Features of R2L sub-minor attack

Features of R2L Sub-Minor attack			
No	**Feature algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f1,f3,f10,f11,f14,f19,f26,f28,f38,f39	74.25%(63561)
2.	Correlation	f33,f39,f19,f38,f14,f10,f22,f36	91.8%(78608)
3.	GainRatio	f11,f14,f15,f9,f18,f10,f19,f13,f17	**95.8(81945)**
4.	InfoGain	f6,f5,f3,f33,f1,f10,f39,f38,f36	89.5%(76610)
5.	OneRAttribute	f6,f5,f1,f10,f39,f3,38	45.2%(38627)
6.	ReliefFAttribut	f3,f33,f36,f34,f32,f12	66.35(56736)
7.	SymmetricalUncert	f10,f39,f38,f3,f1,f19,f14,f11	92.1%(78889)
8.	Average	f1,f3,f5,f6,f10,f11,f14,f19,f33,f36,f38,f39	91.1%(78000)
9.	All Features	f1-f41	89.1 (76280)

7.3.2.3 Ftp Attack vs. Normal Data Experiment

Table 7-10 provides clarification that the various features inherent across the content feature group are supported in terms of managing Ftp attack identification. With the majority of the ranked features identified in this particular group, these elements include f10, f13, f16, f17, f19 and f22, and are referred to as Hot, Num-compromised, Num-root, Num-file-creations, Num-access-files and Is-guest-login, respectively. In line with those feature selection approaches tested, the most commonly ranked features include f9 and f32, which are referred to as Urgent and Dst_host_count, as supported through five of the individual techniques. Nonetheless, such aspects are unique as they have been ranked in consideration to Ftp and Multihop (which will notably undergo examination later on) attack identification. The various traffic features are seen to adopt a key part in the identification of Ftp, with such supported characteristics including f32 (Dst_host_count), f33 (dst_host_srv_count) and f36 (dst_host_same_src_port_rate). Nonetheless, the remaining agreed ranked features fall into the TCP connection feature group, namely F1, f3, f5, f6, and f9.

Table 7-10: Ranked features of Ftp attack

Features of Ftp attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f9,f10,f16,f17,f19,f25,f32,33	99.5%(79953)
2.	Correlation	f9,f19,f22,f36,f33,f37	95.1%(76393)
3.	GainRatio	f9,f13,f17,f10,f19,f22,f32	99.6%(79980)
4.	InfoGain	f5,f6,f3,f33,f32,f36,f1	99.4%(79805)
5.	OneRAttribute	f6,f5,f1,f9,f3,f10,f35	93.3%(74940)
6.	ReliefFAttribut	f3,f36,f33,f32,f2,f34	**99.7%(80111)**
7.	SymmetricalUncert	f9,f32,f19,f17,f22,f16,f13	99.6%(79984)
8.	Average	f1,f3,f5,f6,f9,f10,f13,f16,f17,f19,f22,f32,f33,f36	99.5%(79890)
9.	All features	f1-f41	97.8%(78558)

7.3.2.4 Password Guessing Attack vs. Normal Data Experiment

Those features considered important in the case of password-guessing attacks identification are detailed in the Table 7-11, which shows that the features most commonly supported through the algorithms include f11(num_failed_logins), f39 (dst_host_srv_serror_rate) and f40 (dst_host_rerror_rate), as demonstrated by six of the feature selection methods tested. When identifying such an attack, it is recognised that there are two key aspects, which are concerned with Rerror_rate and Srv_serror_rate, referred to as f40 and f41, respectively.

Table 7-11: Ranked Features of Password Guessing attack

Features of Password Guessing attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f10,f11,39	99.9%(80272)
2.	Correlation	f11,f39,f28,f27,f41,f40,f4	94.7%(76120)
3.	GainRatio	fF11,f39,f10,f38,f4,f40,f41	99.8%(80203)
4.	InfoGain	f5,f11,f6,f4,f3,f10,f39,f38	**99.9%(80316)**
5.	OneRAttribute	f5,f11,f6,f4,f10,f39,f3,f40	**99.9%(80316)**
6.	ReliefFAttribut	f3,f22,f40,f41,f33,f32	99.2%(79696)
7.	SymmetricalUncert	f11,f39,f10,f38,f4,f40	99.8%(80205)
8.	Average	f3,f4,f5,f6,f10,f11,f38,f39,f40,f41	95.8%(77017)
9.	All features	f1-f41	99.16%(79685)

7.3.2.5 Imap Attack vs. and Normal Data Experiment

As shown, Table 7-12, through the results garnered, there are a number of individual aspects that may prove useful in identifying Imap occurrence as not ranked in any other R2l instance under examination. These are recognised as f24 (Srv-count), f25(Serror-rate), f26(Srv-serror-rate), f28 (Srv-rerror-rate) and f31 (Srv_diff_host_rate), with all of these seen to belong to the traffic feature group for the connection between the host and services. Nonetheless, two of the ranked featured aspects are seen to be individual in regards the identification of the Imap and some different R2L sub-minor attack: first, logged_in (f12), which is viewed as being unique in regards the identification of Imap and Warezmaster attacks; and second, lnum_root (f16), which is considered in relation to the identification of Imap, Ftp and Multihop attacks. The remaining ranked characteristics relevant in line with the Imap identification include the following: f3, f5, f33, f36, f38 and f39.

Table 7-12: Ranked features of Imap attack

Features of Imap attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f5,f16,f26,f39	99.4%(79846)
2.	Correlation	f39,f26,f25,f38,f24,f33,f36,f4,f12	**99.9%(80236)**
3.	GainRatio	f26,f39,f38,f25,f16,f13	99.3%(79766)
4.	InfoGain	f3,f39,f38,f26,f25,f24,f5,f31,f28,f33,f6,f36	99.6%(79984)
5.	OneRAttribute	f3,f39,f38,f26,f25,f24,f28,f31	98.5(79473)
6.	ReliefFAttribut	f3,f33,f34,f32,f12,f31	99.8%(80140)
7.	SymmetricalUncert	fF39,f26,f38,f25,f28	99.3%(79767)
8.	Average	f3,f5,f12,f16,f24,f25,f26,f28,f31,f33,f36,f38,f39	99.6%(80014)
9.	All Features	f1-f41	99.5%(79907)

7.3.2.6 Warezmaster Attack vs. Normal Data Experiment

As detailed in the Table 7-13, in the feature ranking carried out by the feature selection algorithms applied, the most common feature is that of dst_host_srv_count, f33, which is known to deal with srv_count across the destination–host link, as supported by the observations of results of six of the different feature selection methods. As has been discussed when examining the Imap attack, imap and warzeclient have the unique common feature of f12, which is concerned with a successful login. The remaining ranked features that play a role in regards to the identification of the warezmaster attack include f1, f3, f5, f6, f10, f17, f22 and f36.

Table 7-13: Ranked features of Warezmaster attack

Features of Warezmaster attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	fF1,f5,f6,f10,f17,f33,f36	98.3%(78976)
2.	Correlation	f6,f36,f17,33,f12,f22,f3	**99.9%(80235)**
3.	GainRatio	f10,f17,f22,f33,f1,f5	95.1%(76374)
4.	InfoGain	f6,f1,f3,f5,f33,f36	98.1%(78839)
5.	OneRAttribute	f6,f1,f17,f10,f5,f38	98.3%(78976)
6.	ReliefFAttribut	f3,f33,f36,f12,f34,f22	99.3%(79801)
7.	SymmetricalUncert	f33,f1,f17,f10,f5,f6,f36	98.3%(78976)
8.	Average	f1,f3,f5,f6,f10,f12,f17,f22,f33,f36	98.7%(79307)
9	All Features	f1-f41	98.3%(78949)

7.2.3.7 Multihop Attack vs. Normal Data Experiment

The most commonly supported feature in the identification of a multi-hop attack is lnum_file_creations, f17, which is observed as ranked by six of the different feature selection methods and is related to file-creation functionality (see Table 7.14). In regards the most common individual features, the unique feature supported in line with the identification of

multi-hop and Ftp is that of lnum_compromised, f13. As has been documented in regards Imap attack analysis, lnum_root (f16) is viewed as being the unique common feature when establishing attacks of a Multihop, Ftp and Imap nature. Furthermore, dst_host_count, f32, which is recognised as managing node calculations across the destination–host journey, is viewed as being a unique characteristic supported in line with the identification of the Multihop and Phf attacks. Furthermore, lnum_shells (f18) is seen to be unique in regards the identification of Multihop and Spy attacks, with the remaining elements warranting attention for multi-hop identification including f1(duration), f3(service), f5 (src_bytes), f6 (dst_bytes), f10 (hot), f14 (lroot_shell), f17 (lnum_file_creations), f22 (is_guest_login), f33 (dst_host_srv_count) and f36(dst_host_same_src_port_rate).

Table 7-14: Ranked features of Multihop attack

Features of Multihop attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f5,f10,f13,f17,f33	95.2%(76434)
2.	Correlation	f14,f18,f16,f17,f13,f6,f22	**99.8%(80121)**
3.	GainRatio	fF16,f13,f14,f18,f17,f10	99.1%(79631)
4.	InfoGain	f6,f5,f1,f33,f10,f17,f36,f32	98.5 (79071)
5.	OneRAttribute	f6,f5,f1,f10,f16,f13,f17,f14,f18	99.5%(79896)
6.	ReliefFAttribut	f3,f33,f36,f12,f34,f22,f32	95.3%(76512)
7.	SymmetricalUncert	fF17,f16,f13,f10,f14,f18	99.1 (79631)
8.	Average	f1,f3,f5,f6,f10,f13,f14,f16,f17,f18,f22,f32,f33,f36	99.5%(79923)
9.	All Features	f1-f41	99.5%(79923)

7.3.2.8 Phf Attack vs. Normal Data Experiment

When seeking to identify attacks of a Phf nature, it is apparent (See Table 7-15) that the implemented feature selection approaches show support for lroot_shell (f14) and lnum_access_files (f19) features. Furthermore, f10, which is commonly referred to as Hot, is commonly supported by all of the six different feature selection algorithms experimented with. Importantly, however, in identifying a PhF, Ftp attack, the f19 feature, known as lnum_access, is observed as a unique common feature. Moreover, in the identification of the Phf, Multihop attacks, lroot_shell (f14) and dst_host_count (f32) are unique. Those ranked features are seen to be common and significant, i.e. f5, f6 and f10, are notably src_bytes, dst_bytes and hot, respectively. Based on the NB detection accuracy of a Phf attack, all of the used feature selection approaches generate a set of features and those features lead to efficient detection of Phf attack. The best set among them is the set that is generated by GainRatio approaches as well as the features in the average set.

Table 7-15: Ranked features of Phf attack

Features of Phf attack			
No	**Feature algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f5,f6,f10,f14,19,f36	99.998%(80314)
2.	Correlation	f14,f19,f10	99.99%(80310)
3.	GainRatio	f14,f10,f19,f6	100%(80316)
4.	InfoGain	f6,f5,f14,f10,f19,f33	99.998%(80314)
5.	OneRAttribute	f5,f6,f14,f10,f19,f1,f28	99.998%(80314)
6.	ReliefFAttribut	f14,f32,f19,f3,f12	99.7%(80071)
7.	SymmetricalUncert	f14,f10,f19,f6,f5	99.998%(80314)
8.	Average	f5,f6,f10,f14,f19,f32	100%
9.	All features	f1-f41	99.85%(80192)

7.3.2.9 Spy Attack vs. Normal Data Experiment

It is observed from the data listed in Table 7-16 that the most commonly ranked feature, as supported by all of the feature selection methods investigated, is dst_host_srv_serror_rate (f39) whereas st_host_serror_rate (f38) was seen to be supported by six of the algorithms. In line with particular features pertaining to the identification of Spy attacks, lsu_attempted (f15) and dst_host_same_srv_rate (f34) are seen to be supported as unique features. As has been discussed previously, in establishing Multi-hop and Spy attacks, a unique feature is that of lnum_shells (f18), which is observed to be unique for the determination of Multihop and Spy attacks. On the other hand, the remaining most commonly ranked features are those of f1, f3, f5, f6, f17, f33 and f34.

Table 7-16: Ranked features of Spy attack

Features of Spy attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f1,f15,f38,f39	99.998%(80312)
2.	Correlation	f39,f15,f18,f38,f19	99.995%(80310)
3.	GainRatio	f39,f38,f15,f18,f17	99.99%(80309)
4.	InfoGain	f39,f38,f1,f6,f33,f34,f3,f5	**100%(80314)**
5.	OneRAttribute	f39,f38,f1,f6,f33,f34,f3,f15,f5,f18	**100%(80314)**
6.	ReliefFAttribut	f3,f12,f34,f33,f32,f39	**100%(80314)**
7.	SymmetricalUncert	f39,f38,f15,f18,f17,f1	99.99%(80209)
8.	Average	f1,f3,f5,f6,f15,f17,f18,f33,f34,f38,f39	**100%(80314)**
9.	All Features	f1-f41	**100%(80314)**

7.3.2.10 Warezclient Attack vs. Normal Data Experiment

It is found that the most commonly ranked feature, as demonstrated by the various methods, include f3 (Service), f5 (src_bytes), f10 (hot) and f22 (is_guest_login), Table 7-17. Moreover, it can be seen that there is a lack of significance in relation to the traffic features group in

regards to the link between host and services for the identification of warezclient attacks, with none of the common ranked features observed to belong to this particular group. Notably, the remaining commonly suggested features for the identification of warezclient occurrence include f1, f6, f10, f33 and f36. However, the outcome of the CFs feature selection algorithm leads to best NB classification accuracy in the classifcation of warezclient attack, where also correlation approach led to closest result to that of the Cfs approach. The features f5, f10, f22 can be observed to be the key features of the warezclient attack classification.

Table 7-17: Ranked Features of the Warezclient attack

Features of warezclient attack			
No	**Feature Algorithms**	**Ranked Features**	**NB**
1.	CFs(BestFirst)	f5,f10,f22	**98.7%(79972)**
2.	Correlation	f10,f22,f5,f33,f36,f3,f37	98.7%(79964)
3.	GainRatio	f22,f10,f3,f5,f33,f6	98.6%(79958)
4.	InfoGain	f5,f3,f6,f33,f10,f36,f22	98.6%(79958)
5.	OneRAttribute	fF5,f10,f6,f22,f3,f1	97.9%(79406)
6.	ReliefFAttribut	f3,f33,f36,f34,f12	96.7%(78434)
7.	SymmetricalUncert	f10,f22,f3,f5,f33,f6,f1	97.7%(79274)
8.	Average	f1,f3,f5,f6,f10,f:22,f33,f36	97.9%(79441)
9.	All features	f1-f41	92.6%(75128)

7.4 Analysis of the High and Low feature ranking results for the minor attacks and their sub-minors attacks based on NB performance

After the specific features set of each minor attacks and their sub-minor is demined in previous Section7.3, the best feature set for each attack is presented in Table 7-18 based on the NB detection accuracy. It can be concluded that the features that are generated by GainRatio algorithm (mostly content features) lead to low false error rate (FR) and best true positive rate (TPR) as the NB based accuracy achieved 97.45%. However, Based on the NB classification accuracies obtained it can be concluded that the Gainratio features lead to the best performance accuracy, approximately 94.9% as 78012 instances are correctly classified. Where It is clearly seen that NB classification for buffer overflow score high accuracy (mostly 99% and above) in all of the used feature selection approaches. The best classifications are reached by correlation approaches as 80183 instances are correctly classified. Based on NB based detection accuracy for the loadmodule attack, the feature sets that are selected by correlation and OneRAttribute approaches leads to the lowest accuracy among the applied approaches. While it is recommended to focus on the features that are selected by ReliefFAttribute algorithm enhance the louadmodule detection as with it, NB scores the highest true classification accuracy of 99.75%. Moreover, it is observed that the features that are agreed by the CFs and

SymmetricalUncert approaches are the best features for the detection of Perl attack, as within it the NB based detection accuracy score is 100%. It is noted also that these features are in the content feature category. However, Rootkit attack detection can be enhanced based on NB, through the focus of the features that are selected by Cfs algorithm since the lowest false detection is scored by the NB through the deployment of those features. In other words, rootkit detection accuracy is the highest when the Cfs feature set is deployed.

In terms of the NB detection for the R2L attack, binary-class classification, the GainRatio generated the best features as it leads to the accuracy of the heights; 98.9%. However, it can be concluded that the Gainratio approach generates the best features that lead to the best detection accuracy of the R2L sub-minor attacks, multi-class classification. Where, among the used feature selection algorithm for FTP, ReleifFattribute leads to best NB detection accuracy. Moreover, InfoGian and OneRAttribute algorithms lead to best detection accuracy for Password Guessing attack among the approaches tested. For the Imap attack, it can be seen that the feature set that is generated by the correlation approach is the best set for Imap classification. Also, the feature set that is selected by the correlation approach contains the best features that allow the NB classifier to classify the warezmaster attack efficiently. From the observed results of multihop attack features, it is concluded that correlation approaches generate the best features for multihop attack classification. In terms of Phf, the best set among them is the set that is generated by GainRatio approaches. For the detection of spy attacks, it is observed that the NB classification accuracy reaches 100% when using most of the feature selection algorithms when all features are applied. It is noted that the feature set that is generated by SymmetricalUncert approaches lead to some misclassifications. For best detection accuracy, the features f5, f10, f22 can be observed to be the key features of the warezclient attack classification.

Table 7-18: High and Low feature ranking results for the minor attacks and their sub-minors attacks based on NB performance

No.	Attack Name	NB Detection Accuracy		FS Algorithms	Ranked Features
1.	U2R	High	97.4%	GainRatio	f14,f13,f17,f10,f9,f18,f11,f16
		Low	90.1%	Cfs (BestFirst)	fF10, f11,f13,f14,f17,f27,f33,f38
2.	U2R Sub-minor	High	94.85%	GainRatio	f18,f14,f17,f9,f13,f16,f11,f10
		Low	14.9%	OneRAttribute	f6,f1,f5,f17,f16,f18,f14,f13,f10
3.	Buffer Overflow	High	99.8%	Correlation	f14,f17,f10,f33,f36,f25,f32,f3
		Low	95.6%	InfoGain	f6,f5,,f3,f10,f1,f14,f13,f33
4.	Loadmodule	High	99.7%	ReliefFAttribut	f3,f33,f32,f35,f34,f36
		Low	95.1%	Correlation	f14,f18,f17,f37,f36,f33
5.	Perl	High	100%	SymmetricalUncert and Cfs (BestFirst)	f1,f14,f16,f17,f18
		Low	99.98%	ReliefFAttribut	f18,f14,f3,f34,f33,f12
6.	Rootkit	High	99.5%	Cfs (BestFirst)	fF11,f13,f16,f17,f33
		Low	92.1%	ReliefFAttribut	f3,f33,f34,f36,f31,f32
7.	R2L attack	High	98.9%	GainRatio	f22,f10,f11,f9,f13,f3
		Low	96.1%	ReliefFAttribut	f14,f3,f32,f33,f2,f36
8.	R2L Sub-minor	High	95.8%	GainRatio	f11,f14,f15,f9,f18,f10,f19,f13,f17
		Low	45.2%	OneRAttribute	f6,f5,f1,f10,f39,f3,38
9.	FTP	High	99.7%	ReliefFAttribut	f3,f36,f33,f32,f2,f34
		Low	93.3%	OneRAttribute	f6,f5,f1,f9,f3,f10,f35
10.	Password Guessing	High	99.9%	InfoGain and OneRAttribute	f5,f11,f6,f4,f10,f39,f3,f40,f38
		Low	94.7%	Correlation	f11,f39,f28,f27,f41,f40,f4
11.	Imap	High	99.9%	Correlation	f39,f26,f25,f38,f24,f33,f36,f4,f12
		Low	98.5%	OneRAttribute	f3,f39,f38,f26,f25,f24,f28,f31
12.	Warezmaster	High	99.9%	Correlation	f6,f36,f17,33,f12,f22,f3
		Low	95.1%	GainRatio	f10,f17,f22,f33,f1,f5
13.	Multihop	High	99.8%	Correlation	f14,f18,f16,f17,f13,f6,f22
		Low	95.2%	CFs(BestFirst)	f5,f10,f13,f17,f33
14.	Phf	High	100%	GainRatio	f14,f10,f19,f6
		Low	99.7%	ReliefFAttribut	f14,f32,f19,f3,f12
15.	Spy	High	100%	InfoGain, OneRAttribute and ReliefFAttribut	f39,f38,f1,f6,f33,f34,f3,f15,f5,f18 ,f12,f32
		Low	99.99%	SymmetricalUncert	f39,f38,f15,f18,f17,f1
16.	Warezclient	High	98.7%	CFs(BestFirst)	f5,f10,f22
		Low	96.7%	ReliefFAttribut	f3,f33,f36,f34,f12

7.5 Summary and Conclusion

Throughout the course of this chapter, a rigorous investigation has been carried out as to the classification of two minor attacks and their corresponding sub-minor attacks, utilising seven feature ranking, feature selection algorithms. The experiments have been separated into two groups: in the preliminary instance, the U2R attack feature and its corresponding sub-minor attacks were analysed; in the second instance, the emphasis was placed on R2L attacks and their sub-minor attacks. Across both of these stages, two different datasets are created and applied on a binary-class dataset or a multi-class dataset.

Examining the U2R attacks and their sub-minor attacks emphasised that those features in the first, second and third feature groups within the KDD cup dataset are recognised as the most significant when it comes to identifying the U2R attacks, owing to the fact that the majority of the ranked features supported by the observations of the ranked feature list that belongs to such groups. Importantly, semantic data, which is seen to be difficult to capture across the initial phases, is necessary when examining attacks that are U2R in nature. Such an attack is most predominantly content-based. Owing to the fact that U2R attacks present an outcome that is 'root-shell' obtained without legitimate means and root-relevant characteristics appear to be valuable in the identification, the feature selection methods experimented with have opted to incorporate various root-relevant features. As such, upon there being an attack of a U2R nature, a number of different features, including the number of instances of shell prompts or the number of file creations, are selected; on the other hand, other features are neglected, as in the cases of protocol and source bytes, for example.

When seeking to establish identification, R2l attacks are viewed as being the most troublesome, predominantly considering their involvement of network- and host-level characteristics. As such, both host- and network-level components-namely 'duration of connection' and 'service requested', and the 'number of failed login attempts', respectively-are selected in the establishment of R2L attacks. In considering the form of operation manipulated by the R2L attacks and corresponding sub-minors, the significant features when it comes to recognising such forms are seen to belong to the majority of the feature groups in the KDD cup dataset. Importantly, most IMAP applications present the potential of various logins, which fundamentally positions the end user in such a way that they are able to link to the email server through various instruments simultaneously. As such, traffic features need to be determined through the presence of a host–services connection incorporating a two-second time window.

In line with the Ftp exploitation tool, which is generalised as an attack on the FTP protocol through the attacker making use of the PORT command with the aim of achieving access to ports, this is commonly recognised as an Ftp bounce attack. Accordingly, in Ftp identification, the aspects of urgent, Num-compromised, Num-access-files and Dst-host-count are essential, as determined through the methods as being unique features for Ftp. In regards to the password-guessing form of attack, there are also a number of different features chosen by the algorithms tested, including Num-failed-logins, Dst-host-srv-serror-rate, and Dst-host-error-rate. Owing to the fact that such an attack arises following various efforts being made to log in, it is possible

to establish the number of failed login attempts, in addition to other features. Through a multi-hop attack, it is possible for a particular attack path or chin to be followed by an attacker, notably with the objective to achieve access to the target. As such, the most significant features recognised as unique features in the number of compromised (number of file/path not found errors and jumping commands) conditions, the number of shells prompts, and those that establish the Count traffic in the destination–host link, is critical.

Nonetheless, there is the uploading and downloading of data from various hidden directions when there is an FTP connection, which subsequently constitutes warezmaster and warezclient R2l attacks. With this in mind, the number of significant features deemed pertinent to achieving the most optimal attack outcomes is determined; it may also be seen that there are a number of different characteristics that are specific to warezmaster, which are logged_in and lnum_file_creations. Owing to the fact that the warezmaster mechanism is highlighted in such a way through the granting of write permission to a user, it is then possible that the attacker can gain access to the server by logging in using anonymous credentials; this, in turn, highlights the potential for directories and files to be uploaded. In this case, it is possible for warezclient to download malicious files uploaded to the server by the attacker, with any anonymous/legal user able to do so.

In general the rigorous analysis that was carried out in this chapter with regards to the determination of a reduced set of most significant features that will enable the optimisation of the NB classifier performance provides us with additional knowledge as to the best set of features, in their rank order, can be used to most accurately classify a given attack.

Chapter 8 : Summary, Conclusion and Future Work

8.1 Summary and Conclusion

The research conducted in this thesis initially provided an explanation pertaining to the key concepts of Cloud Computing and further provided a literature review with the aim of examining the cloud's security gaps. There are a number of different gaps in the security of Cloud Computing, which notably impacts all of its layers. In order to narrow down the scope of this research study, a decision was made to focus on the IaaS layer; a part of the network security layer. The choice of IaaS can be explained owing to the fact that all layers beside are built on top of it. Importantly, the decision to focus on network security was based on its practical importance as a fundamental aspect of the Cloud, meaning that any weaknesses present in the network have a profound and direct impact on the overall security of the Cloud.

As has been highlighted throughout the literature, a number of different intrusions and attacks can impact overall network security, meaning Cloud network security is enhanced via the adoption of more commonplace defence approaches, including IDSs and firewalls, for example. With this noted, and in an effort to further reduce the scope of this work, the decision was made to utilise IDSs owing to the inability of firewalls to identify complex attacks, such as those of DoS and DDoS, and also insider attacks. In considering the aim of the research conducted in this thesis focused on improving the security of the Cloud via the application of IDSs, a review pertaining to the use of Artificial Intelligence (AI) has been carried out in line with intrusion detection in order to build an intelligent system that has the ability to improve IDS performance. Importantly, however, when seeking to examine the performance of the IDS, different datasets had to be used as tools of comparison.

Following a number of different considerations, the KDD Cup 99 dataset was selected as the key dataset to be used owing to the recognition that the limitations inherent in other datasets will not be an issue in this dataset. Notably, however, there are limitations inherent in this dataset, including the fact that some machine-learning classifiers may demonstrate inadequate performance in the current specific features of this selected dataset. The most widely influenced in this regard is the fact that the KDD Cup 99 is an imbalanced dataset, which arises from the fact that there is a significant outnumber of instances in one class over the numbers of another. Such an imbalanced dataset may create significant challenges in attack classification, either at the multi-class or binary class level, with both demonstrating some degree of bias in relation to the major classes and, as a result, leading to minor class misclassifications. The majority of

researchers in this regard have centred their attention on binary class misclassifications only; this is predominantly owing to the complexity of the multi-class classification, which warrants the study of different classes.

The Chapter 4 directed attention to the multi-class problem on KDD Cup ’99 imbalanced dataset, meaning that the imbalance dataset is explored in relation to the approaches that are applied in order to improve machine learning classifier performance. Following the conduct of research and rigorous examination of results in this regards, it has been concluded that a number of classifiers show bias towards major classes—which can subsequently result in minor class misclassification at times, or the minor classes being completely disregarded. As explained throughout the literature, there is the view that there are only three key approaches to be learned with regards to imbalanced data: data level, algorithm level method, and ensemble methods. The experiments completed in Chapter 4 recommended the use of cross-validation to avoid the diversity of the results. It was also concluded that the imbalanced dataset is recognised as the reason behind the inadequate performance of some classifiers.

The way in which the Naives Byase classifier operates in the case of imbalanced data has been examined in Chapter 5, with the conclusion drawn that for NB based network attack identification in the case of minor attacks, i.e. R2L and U2R, requires improvement via the application of a number of different imbalanced learning methods. Nonetheless, following the application of resampling, U2R identification is essentially enhanced when compared with R2L. Following an in-depth exploration, it was stated that, as a result of various misclassifications, R2L identification has not achieved any improvement, despite the adoption of bagging and resampling. Accordingly, the research directs its attention towards the R2L sub-minor attacks, which established that inadequate levels of NB detection is owing to Multihop and Warezclient attacks: these were incorrectly classified as a result of features/behaviour of some of the feature selection approaches, applied.

It was shown that in the KDD cup’ 99 dataset, class distributions are not balanced and there is also the need to reduce the dimensionality of the feature space. Chapter 6 investigated whether feature selection followed by a method combating the skewness of the dataset or the opposing pipelines will perform better. The statistical analysis of the results revealed that both of the pipelines should be considered when training for the best classification models; in particular, it is based on the classifier used and the feature selection method utilised. In more of the cases, feature selection after resampling approach outperformed the opposing pipeline.

However, it was observed that each sub-minor attacks have their own mechanism to exploit, each of sub-minor attack has their own selected features some of them are common with the other attacks while some of it is unique.

8.2 Future Work

Although this thesis carried out a comprehensive and rigorous study of the potential to use machine learning algorithms in the recognition of attacks in an imbalanced IDS related dataset, further research could have been conducted, time permitted. Below is a list of further directions of research that is recommended based on the research findings of this thesis:

- It will be possible to fine-tune the feature selection algorithms utilised in this research by carrying out a sensitivity analysis of their parameters with the view to selecting the optimal set of parameters that will result eventually on optimisation of the classification accuracies.
- The research conducted within the context of this thesis can be verified, further supported and enhanced through the application of deep learning.
- The feature selection impact should be investigated via the adoption of other commonly used classifiers, such as random forest, for example.
- Owing to a lack of Cloud-specific, attack datasets, there is a need to collect further useful data and making them publically available.
- Chapter-7 conducted a comprehensive and rigorous study of the significance of attack features/parameters on the attack classification accuracy when different, popular classification approaches are used. For this purpose feature selection algorithms were used which provided the features in their rank order. The Rank order thus obtained provides vital features of an attack that could be used in understanding the unique characteristics of different kinds of attacks, leading to the possibility that this subject understanding can lead to interesting findings on how best to design a software system that will most efficient in the detection of network intrusions.

References

[1] Cloud Security Alliance, “The Treacherous 12 - Top Threats to Cloud Computing + Industry Insights,” *Cloud Secur. Alliance*, p. 60, 2017.

[2] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, “A survey on security issues and solutions at different layers of Cloud computing,” *J. Supercomput.*, vol. 63, no. 2, pp. 561–592, 2013.

[3] Y. Chen and R. Sion, “On Securing Untrusted Clouds with Cryptography,” *Science (80-.).*, pp. 109–114, 2010.

[4] V. Engen, “Machine Learning For Network Based Intrusion Detection,” *Int. J.*, 2010.

[5] M. Galar, A. Fern, E. Barrenechea, and H. Bustince, “Hybrid-Based Approaches,” vol. 42, no. 4, pp. 463–484, 2012.

[6] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 4, pp. 463–484, 2012.

[7] M. Vouk *et al.*, “‘ Powered by VCL ’ - Using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing,” *East*, vol. 6, no. Vcl, pp. 1–10, 2008.

[8] M. Armbrust *et al.*, “A view of cloud computing,” *Commun. ACM*, vol. 53, no. 4, p. 50, 2010.

[9] M. Armbrust, a Fox, R. Griffith, A. Joseph, and Rh, “Above the clouds: A Berkeley view of cloud computing,” *Univ. California, Berkeley, Tech. Rep. UCB* , pp. 07–013, 2009.

[10] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: State-of-the-art and research challenges,” *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, 2010.

[11] P. G. Tim Mell, “Draft NIST Working Definition of Cloud Computing,” *Natl. Inst. Stand. Technol.*, vol. 53, p. 50, 2009.

[12] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming, 1st edition*. 2013.

[13] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, “Cloud computing: Opportunities and challenges,” *IEEE Potentials*, vol. 33, no. 1, pp. 34–36, 2014.

[14] Z. Erl, T., Puttini, R. & Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, vol. 1. Prentice Hall PTR., 2015.

[15] N. Leavitt, “Is cloud computing really ready for prime time?,” *Comput. Soc. IEEE*, vol. 42, no. 1, pp. 15–25, 2009.

[16] Microsoft TechNet, “Common Types of Network Attacks,” *Microsoft Technet*, vol. 959354. pp. 1–3, 2011.

[17] wikipedia, “Attack (computing) - Wikipedia, the free encyclopedia.” 2015.

[18] V. Cerf, H. Kong, Y. Dalal, C. Sunshine, and P. R. Net, “Denial-of-service attack - Wikipedia, the free encyclopedia,” *North*, 2009. [Online]. Available: http://en.wikipedia.org/wiki/California_Eagle.

[19] Wikipedia, “Denial-of-service attack - Wikipedia.” .

[20] “Article_ K14813 - Detecting and mitigating DoS_DDoS attacks (11.” .

[21] N. Gruschka and M. Jensen, “Attack surfaces: A taxonomy for attacks on cloud services,” *Proc. - 2010 IEEE 3rd Int. Conf. Cloud Comput. CLOUD 2010*, pp. 276–279, 2010.

[22] D. D. I. Informatica, D. Di, R. In, and P. H. D. Thesis, “Cloud Computing Security , An Intrusion Detection System for Cloud Computing Systems Hesham Abdelazim Ismail Mohamed To the most precious inspiration of my life : My parents and my brothers and sisters,” 2013.

[23] C. N. Modi, D. R. Patel, A. Patel, and M. Rajarajan, “Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing,” *Procedia Technol.*, vol. 6, pp. 905–912, 2012.

[24] K. Patel and R. Srivastava, “Classification of Cloud Data using Bayesian Classification,” *Int. J. Sci. Res.*, vol. 2, no. 6, pp. 2–7, 2013.

[25] R. Bace, “NIST special publication on intrusion detection systems,” *Nist Spec. Publ.*, pp. 1–51, 2001.

[26] T. W. Purboyo, B. Rahardjo, and Kuspriyanto, “Security metrics: A brief survey,” *2011 Int. Conf. Instrumentation, Commun. Inf. Technol. Biomed. Eng.*, no. November, p. 4, 2011.

[27] R. M. Savola, “A Security Metrics Taxonomization Model for Software-Intensive Systems,” *J. Inf. Process. Syst.*, vol. 5, no. 4, pp. 197–206, 2009.

[28] J. Arshad, P. Townend, and J. Xu, “A novel intrusion severity analysis approach for Clouds,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 416–428, 2013.

[29] P. a Porras and P. G. Neumann, “EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances,” *Proc. 20th NIST-{NCSC} Natl. Inf. Syst. Secur. Conf.*, pp. 353–365, 1997.

[30] H. a. Kholidy and F. Baiardi, “CIDS: A framework for intrusion detection in cloud systems,” *Proc. 9th Int. Conf. Inf. Technol. ITNG 2012*, pp. 379–385, 2012.

[31] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, and A. Patel, “An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2105–2127, 2013.

[32] H. T. Elshoush and I. M. Osman, “Alert correlation in collaborative intelligent intrusion detection systems - A survey,” *Appl. Soft Comput. J.*, vol. 11, no. 7, pp. 4349–4365, 2011.

[33] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, “The 1999 DARPA o € -line intrusion detection evaluation,” *Comput. Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[34] S. Roschke, F. Cheng, and C. Meinel, “Intrusion detection in the cloud,” *8th IEEE Int. Symp. Dependable, Auton. Secur. Comput. DASC 2009*, pp. 729–734, 2009.

[35] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology,” *Nist Spec. Publ.*, vol. 800, p. 94, 2007.

[36] S. Roschke, F. Cheng, and C. Meinel, “An extensible and virtualization-compatible IDS management architecture,” *5th Int. Conf. Inf. Assur. Secur. IAS 2009*, vol. 2, pp. 130–134, 2009.

[37] B. Tjaden *et al.*, “INBOUNDS: The Integrated Network-Based Ohio University Network Detective,” 2000.

[38] M. Dacier and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” 1999.

[39] P. Helman and G. Liepins, “Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse,” *IEEE Trans. Softw. Eng.*, vol. 19, no. 9, pp. 886–901, 1993.

[40] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” *Comput. Networks*, vol. 31, pp. 805–822, 1999.

[41] T. M. Mitchell, “The Discipline of Machine Learning,” *Mach. Learn.*, vol. 17, no. July, pp. 1–7, 2006.

[42] R. Grossman, S. Kasif, R. Moore, D. Rocke, and J. Ullman, “Data Mining Research: Opportunities and Challenges,” vol. 1998, 1999.

[43] S. P. Portillo, “PhD Thesis Attacks Against Intrusion Detection Networks : Evasion , Reverse Engineering and Optimal Countermeasures,” no. June, 2014.

[44] L. Long, X. Wang, and X. Zhu, “Machine Learning in Network Intrusion Detection,” vol. 11, no. 2, p. 9941, 2015.

[45] R. Sommer and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” pp. 305–316, 2010.

[46] S. N. S. Naiping and Z. G. Z. Genyuan, “A Study on Intrusion Detection Based on Data Mining,” *Inf. Sci. Manag. Eng. ISME 2010 Int. Conf.*, vol. 1, pp. 8–15, 2010.

[47] C. Thomas and N. Balakrishnan, “Performance enhancement of Intrusion Detection Systems using advances in sensor fusion,” *Inf. Fusion, 2008 11th Int. Conf.*, pp. 1–7, 2008.

[48] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, pp. 1–6, 2009.

[49] J. McHugh, “Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory,”

ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp. 262–294, 2000.

[50] C. Thomas and N. Balakrishnan, "Performance enhancement of Intrusion Detection Systems using advances in sensor fusion," *2008 11th Int. Conf. Inf. Fusion*, pp. 1–7, 2008.

[51] M. Tavallaee, "An Adaptive Intrusion Detection System," *Sdstate.Edu*, 2011.

[52] V. Engen, J. Vincent, and K. Phalp, "Exploring discrepancies in findings obtained with the KDD Cup '99 data set," *Intell. Data Anal.*, vol. 15, no. 2, pp. 251–276, 2011.

[53] M. M. Andreasen, C. T. Hansen, and P. Cash, "Good Design," *Concept. Des.*, vol. 45, no. 21, pp. 369–389, 2015.

[54] M. Jouini, L. B. A. Rabai, and A. Ben Aissa, "Classification of security threats in information systems," *Procedia Comput. Sci.*, vol. 32, pp. 489–496, 2014.

[55] S. S. Kaushik and P. R. Deshmukh, "Detection of Attacks in an Intrusion Detection System," *Int. J. Comput. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 982–986, 2011.

[56] J. Postel and J. Reynolds, "FILE TRANSFER PROTOCOL (FTP)," 2011. .

[57] M. Rouse, "What is IMAP (Internet Message Access Protocol)? - Definition from WhatIs.com." 2015.

[58] D. Dey, A. DInda, P. P. Kundapur, and R. Smitha, "Warezmaster and Warezclient: An implementation of FTP based R2L attacks," *8th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2017*, pp. 6–11, 2017.

[59] S. Akasapu, "An Integrated Approach for detecting DDoS attacks in Cloud Computing," no. June, pp. 258–261, 2017.

[60] A. S. Janusz S. Kowalik, Janusz Gorski, *Cyberspace Security and Defense: Research Issues*. 2006.

[61] VERACODE, "Rootkit: What is a Rootkit and How to Detect It | Veracode." 2017.

[62] T. Tran, P. Tsai, T. Jan, and X. Kong, "Network Intrusion Detection using Machine Learning and Voting techniques," *Mach. Learn.*, pp. 7–10, 2010.

[63] C. hui Tsai, L. chiu Chang, and H. cherng Chiang, "Forecasting of ozone episode days

by cost-sensitive neural network methods," *Sci. Total Environ.*, vol. 407, no. 6, pp. 2124–2135, 2009.

[64] M. Troesch and I. Walsh, "Machine Learning for Network Intrusion Detection," pp. 1–5, 2014.

[65] S. Juma, Z. Muda, M. M.A., and W. Yassin, "Machine Learning Techniques for Intrusion Detection System: A Review," *J. Theor. Appl. Inf. Theory*, vol. 72, no. 3, pp. 422–429, 2015.

[66] M. Panda, A. Abraham, S. Das, and M. R. Patra, "Network intrusion detection system: A machine learning approach," *Intell. Decis. Technol.*, vol. 5, no. 4, pp. 347–356, 2011.

[67] M. Kubat, "Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7.," *The Knowledge Engineering Review*, vol. 13, no. 4. pp. 409–412, 1999.

[68] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. M??ller, "Efficient backprop," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 9–48, 2012.

[69] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci. (Ny).*, vol. 250, pp. 113–141, 2013.

[70] N. V Chawla, "Data Mining for Imbalanced Datasets: An Overview," *Data Min. Knowl. Discov. Handb.*, pp. 853–867, 2005.

[71] G. M. Weiss, "Mining with Rarity: A Unifying Framework," *SIGKDD Explor.*, vol. 6, no. 1, pp. 7–19, 2004.

[72] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets : A review," *Science (80-.).*, vol. 30, no. 1, pp. 25–36, 2006.

[73] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. Artif. Intell. Res.*, vol. 19, pp. 315–354, 2003.

[74] A. H. R. Ko, R. Sabourin, and A. S. Britto, "From dynamic classifier selection to

dynamic ensemble selection," *Pattern Recognit.*, vol. 41, no. 5, pp. 1735–1748, 2008.

[75] R. Batuwita and V. Palade, "microPred: Effective classification of pre-miRNAs for human miRNA gene prediction," *Bioinformatics*, vol. 25, no. 8, pp. 989–995, 2009.

[76] H.-Y. Lo *et al.*, "Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method," *ACM SIGKDD Explor. Newsl.*, vol. 10, no. 2, p. 43, 2008.

[77] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from imbalanced data in surveillance of nosocomial infection," *Artif. Intell. Med.*, vol. 37, no. 1, pp. 7–18, 2006.

[78] L. Mena and J. a Gonzalez, "Machine learning for imbalanced datasets: Application in medical diagnostic," *Breast*, pp. 574–579, 2006.

[79] A. Al-Shahib, R. Breitling, and D. R. Gilbert, "Predicting protein function by machine learning on amino acid sequences--a critical evaluation.," *BMC Genomics*, vol. 8, p. 78, 2007.

[80] Ł. Kobyliński and A. Przepiórkowski, "Definition extraction with balanced random forests," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5221 LNAI, pp. 237–247, 2008.

[81] K. Kermanidis, M. Maragoudakis, N. Fakotakis, and G. Kokkinakis, "Learning Greek Verb Complements : Addressing the Class Imbalance," no. Laurikkala 2001, 2002.

[82] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem," *Inf. Process. Manag.*, vol. 44, no. 2, pp. 790–799, 2008.

[83] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets," *2006 IEEE Int. Conf. Granul. Comput.*, pp. 732–737, 2006.

[84] R. Barandela, J. S. Sanchez, V. Garcia, and E. Rangel, "Strategies for learning in class imbalance problems.pdf," *Pattern Recog.*, vol. 36, pp. 849–851, 2003.

[85] P. Ducange, B. Lazzerini, and F. Marcelloni, "Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets," *Soft Comput.*, vol. 14, no. 7, pp. 713–728, 2010.

[86] W. J. Lin and J. J. Chen, “Class-imbalanced classifiers for high-dimensional data,” *Brief. Bioinform.*, vol. 14, no. 1, pp. 13–26, 2013.

[87] J. Wang, J. You, Q. Li, and Y. Xu, “Extract minimum positive and maximum negative features for imbalanced binary classification,” *Pattern Recognit.*, vol. 45, no. 3, pp. 1136–1145, 2012.

[88] R. Batuwita and V. Palade, “Class Imbalance Learning Methods for Support Vector,” *Imbalanced Learn. Found. Algorithms, Appl.*, pp. 83–100, 2013.

[89] N. García-Pedrajas, J. Pérez-Rodríguez, M. García-Pedrajas, D. Ortiz-Boyer, and C. Fyfe, “Class imbalance methods for translation initiation site recognition in DNA sequences,” *Knowledge-Based Syst.*, vol. 25, no. 1, pp. 22–34, 2012.

[90] P. Domingos, “MetaCost: A General Method for Making Classifiers Cost-Sensitive,” *Proc. fifth ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, vol. 55, pp. 155–164, 1999.

[91] Z. H. Zhou and X. Y. Liu, “Training cost-sensitive neural networks with methods addressing the class imbalance problem,” *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, 2006.

[92] J. Błaszczyński, M. Deckert, J. Stefanowski, and S. Wilk, “Integrating selective pre-processing of imbalanced data with Ivotes ensemble,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6086 LNAI, pp. 148–157, 2010.

[93] N. V Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, “SMOTEBoost : Improving Prediction,” *Lect. Notes Comput. Sci.*, vol. 2838, pp. 107–119, 2003.

[94] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “RUSBoost: A hybrid approach to alleviating class imbalance,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[95] R. Batuwita and V. Palade, “Efficient resampling methods for training support vector machines with imbalanced datasets,” *Proc. Int. Jt. Conf. Neural Networks*, 2010.

[96] A. Fernández, M. J. del Jesus, and F. Herrera, “On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets,” *Inf.*

Sci. (Ny)., vol. 180, no. 8, pp. 1268–1291, 2010.

[97] A. Fernandez, S. Garc??a, M. J. del Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets Syst.*, vol. 159, no. 18, pp. 2378–2398, 2008.

[98] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," *Proc. 2000 Int. Conf. Artif. Intell.*, pp. 111--117, 2000.

[99] J. Van Hulse, "An Empirical Comparison of Repetitive Undersampling Techniques," pp. 29–34, 2009.

[100] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[101] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 4626–4636, 2009.

[102] A. S. Nickerson, N. Japkowicz, and E. Milios, "Using Unsupervised Learning to Guide Resampling in Imbalanced Data Sets," *Proc. Eighth Int. Work. AI Statitsics*, p. 5, 2001.

[103] A. Estabrooks and N. Japkowicz, "A mixture-of-experts framework for learning from imbalanced data sets," *Adv. Intell. Data Anal.*, pp. 34–43, 2001.

[104] K. Yoon and S. Kwek, "An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics," *Proc. - HIS 2005 Fifth Int. Conf. Hybrid Intell. Syst.*, vol. 2005, pp. 303–308, 2005.

[105] S. J. Yen and Y. S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," *Lect. Notes Control Inf. Sci.*, vol. 344, pp. 731–740, 2006.

[106] Q. Wang, "A Hybrid Sampling SVM Approach to," vol. 2014.

[107] N. V. Chawla, "C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure," *Proc. Int. Conf. Mach. Learn. Work. Learn. from Imbalanced Data Set II*, 2003.

[108] S. Choudhury and A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," *2015 Int. Conf. Smart Technol.*

Manag. Comput. Commun. Control. Energy Mater., no. May, pp. 89–95, 2015.

[109] G. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?," *Dmin*, pp. 1–7, 2007.

[110] D. Adamu Teshome and V. S. Rao, "A Cost Sensitive Machine Learning Approach for Intrusion Detection," *Glob. J. Comput. Sci. Technol.*, vol. 14, no. 6, 2014.

[111] J. Liu, Q. Hu, and D. Yu, "A weighted rough set based method developed for class imbalance learning," *Inf. Sci. (Ny).*, vol. 178, no. 4, pp. 1235–1256, 2008.

[112] N. V Chawla, "Data Mining for Imbalanced Datasets: An Overview," *Data Min. Knowl. Discov. Handb.*, no. January 2005, pp. 853–867, 2005.

[113] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[114] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," *Proc. 24th Int. Conf. Mach. Learn.*, pp. 935–942, 2007.

[115] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory Undersampling for Class Imbalance Learning," *IEEE Trans. Syst. Man Cybern.*, vol. 39, no. 2, pp. 539–550, 2009.

[116] S. Wang and X. Yao, "Relationships between diversity of classification ensembles and single-class performance measures," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 206–219, 2013.

[117] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "2010-IEEE TSMCpartA-RUSBoost A Hybrid Approach to Alleviating Class Imbalance.pdf," vol. 40, no. 1, p. 13, 2010.

[118] R. Barandela, J. S. Sánchez, and R. M. Valdovinos, "New Applications of Ensembles of Classifiers," *Pattern Anal. Appl.*, vol. 6, no. 3, pp. 245–256, 2003.

[119] F. Morstatter and Z. Zheng, "Advancing Feature Selection Research − ASU Feature Selection Repository," 2010.

[120] B. L. Guedes, E. B. Orzolin, and K. D. Keller, "Selection of Relevant Features in Machine Learning," *AAAI Tech. Rep. .*, pp. 127–131, 2014.

[121] G. H. John, R. Kohavi, and Karl Pfleger, “Irrelevant Features and the Subset Selection Problem,” *Icml*, pp. 121–129, 1994.

[122] Z. L. Sun, D. S. Huang, and Y. M. Cheun, “Extracting nonlinear features for multispectral images by FCMC and KPCA,” *Digit. Signal Process. A Rev. J.*, vol. 15, no. 4, pp. 331–346, 2005.

[123] J. L. Crowley and A. C. Parker, “A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 2, pp. 156–170, 1984.

[124] Z. L. Sun, D. S. Huang, Y. M. Cheung, J. Liu, and G. Bin Huang, “Using FCMC, FVS, and PCA techniques for feature extraction of multispectral images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 2, pp. 108–112, 2005.

[125] A. Khotanzad and Y. H. Hong, “Rotation invariant image recognition using features selected via a systematic method,” *Pattern Recognit.*, vol. 23, no. 10, pp. 1089–1101, 1990.

[126] N. Vasconcelos, “Feature selection by maximum marginal diversity: optimality and implications for visual recognition,” *2003 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2003. Proceedings.*, vol. 1, 2003.

[127] N. Vasconcelos, “Scalable Discriminant Feature Selection for Image Retrieval and Recognition 2 . Information theoretic feature selec- tion,” *Computer (Long. Beach. Calif).*, pp. 0–5, 2004.

[128] J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, “Boosting color feature selection for color face recognition,” *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1425–1434, 2011.

[129] A. Goltsev and V. Gritsenko, “Investigation of efficient features for image recognition by neural networks,” *Neural Networks*, vol. 28, pp. 15–23, 2012.

[130] D. L. Swets and J. J. Weng, “Efficient Content-Based Image Retrieval using Automatic Feature Select,” pp. 85–90, 1995.

[131] D. L. Swets and J. J. Weng, “Using Discriminant Eigenfeatures for Image Retrieva,” vol. 18, no. 8, pp. 831–836, 2001.

[132] L. Samiolo, M. Valigi, D. Gazzoli, and R. Amadelli, “Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection,” *Electrochim. Acta*, vol. 55, no. 26, pp. 7788–7795, 2010.

[133] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, “Mutual information-based feature selection for intrusion detection systems,” *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1184–1199, 2011.

[134] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, “Using feature selection for intrusion detection system,” *2012 Int. Symp. Commun. Inf. Technol. Isc. 2012*, pp. 296–301, 2012.

[135] D. D. Lewis, Y. M. Yang, T. G. Rose, and F. Li, “RCV1: A new benchmark collection for text categorization research,” *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

[136] H.-B. S. LI-PING. JING, HOU-KUAN. HUANG, “IMPROVED FEATURE SELECTION APPROACH TFIDF IN TEXT MINING,” no. November, pp. 4–5, 2002.

[137] S. Van Landeghem, T. Abeel, Y. Saeys, and Y. Van De Peer, “Discriminative and informative features for biomolecular text mining with ensemble feature selection,” *Bioinformatics*, vol. 27, no. 13, pp. i554–i560, 2011.

[138] Q. Song, J. Ni, and G. Wang, “A fast clustering-based feature subset selection algorithm for high-dimensional data,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, 2013.

[139] G. Li, X. Hu, X. Shen, X. Chen, and Z. Li, “A novel unsupervised feature selection method for bioinformatics data sets through feature clustering,” *2008 IEEE Int. Conf. Granul. Comput. GRC 2008*, pp. 41–47, 2008.

[140] Y. F. Gao, B. Q. Li, Y. D. Cai, K. Y. Feng, Z. D. Li, and Y. Jiang, “Prediction of active sites of enzymes by maximum relevance minimum redundancy (mRMR) feature selection,” *Mol. Biosyst.*, vol. 9, no. 1, pp. 61–69, 2013.

[141] D. S. Huang and C. H. Zheng, “Independent component analysis-based penalized discriminant method for tumor classification using gene expression data,” *Bioinformatics*, vol. 22, no. 15, pp. 1855–1862, 2006.

[142] C. Zheng, D. Huang, S. Member, L. Zhang, and X. Kong, “Tumor Clustering Using Nonnegative Matrix,” *Technology*, vol. 13, no. 4, pp. 599–607, 2009.

[143] D. S. Huang and H. J. Yu, “Normalized feature vectors: A novel alignment-free sequence comparison method based on the numbers of adjacent amino acids,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 10, no. 2, pp. 457–467, 2013.

[144] L. Wang and J. S. Yu, “Fault feature selection based on modified binary PSO with mutation and its application in chemical process fault diagnosis,” *Adv. Nat. Comput. Pt 3, Proc.*, vol. 3612, pp. 832–840, 2005.

[145] S. Chebrolu, A. Abraham, and J. P. Thomas, “Feature deduction and ensemble design of intrusion detection systems,” *Comput. Secur.*, vol. 24, no. 4, pp. 295–307, 2005.

[146] K. Zhang, Y. Li, P. Scarf, and A. Ball, “Feature selection for high-dimensional machinery fault diagnosis data using multiple models and Radial Basis Function networks,” *Neurocomputing*, vol. 74, no. 17, pp. 2941–2952, 2011.

[147] H. Li, C. J. Li, X. J. Wu, and J. Sun, “Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine,” *Appl. Soft Comput. J.*, vol. 19, pp. 57–67, 2014.

[148] T. Khoshgoftaar, D. Dittman, R. Wald, and A. Fazelpour, “First order statistics based feature selection: A diverse and powerful family of feature seleciton techniques,” *Proc. - 2012 11th Int. Conf. Mach. Learn. Appl. ICMLA 2012*, vol. 2, pp. 151–157, 2012.

[149] J. Gibert, E. Valveny, and H. Bunke, “Feature selection on node statistics based embedding of graphs,” *Pattern Recognit. Lett.*, vol. 33, no. 15, pp. 1980–1990, 2012.

[150] M. C. Lane, B. Xue, I. Liu, and M. Zhang, “Gaussian Based Particle Swarm Optimisation and Statistical Clustering for Feature Selection,” pp. 133–144, 2014.

[151] L. Shen and L. Bai, “Information theory for gabor feature selection for face recognition,” *EURASIP J. Appl. Signal Processing*, vol. 2006, pp. 1–11, 2006.

[152] B. Morgan, *Model Selection and Multimodel Inference :A Practical Information-Theoretic Approach Second Edition*. 2001.

[153] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: criteria of max-dependency,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[154] F. Fleuret, “Fast binary feature selection with conditional mutual information,” *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, 2004.

[155] H. Yang and J. Moody, “Data Visualization and Feature Selection : New Algorithms for Nongaussian Data,” *Neural Inf. Process. Syst.*, no. Mi, pp. 687--693, 1999.

[156] B. I. Bonev, “Feature Selection based on Information Theory Boyán,” pp. 1–10, 2010.

[157] Z. Xu, I. King, M. R. T. Lyu, and R. Jin, “Discriminative semi-supervised feature selection via manifold regularization,” *IEEE Trans. Neural Networks*, vol. 21, no. 7, pp. 1033–1047, 2010.

[158] B. Jie, D. Zhang, B. Cheng, and D. Shen, “chp2F978-3-642-40811-3_35.pdf,” pp. 275–283, 2013.

[159] B. Li, C. H. Zheng, and D. S. Huang, “Locally linear discriminant embedding: An efficient method for face recognition,” *Pattern Recognit.*, vol. 41, no. 12, pp. 3813–3821, 2008.

[160] R. W. Swiniarski and A. Skowron, “Rough set methods in feature selection and recognition,” *Pattern Recognit. Lett.*, vol. 24, no. 6, pp. 833–849, 2003.

[161] J. Wu, T. Qiu, L. Wang, and H. Huang, “An Approach to Feature Selection Based on Ant Colony Optimization and Rough Set,” *Commun. Comput. Inf. Sci.*, vol. 134, no. PART 1, pp. 466–471, 2011.

[162] W. Shu and H. Shen, “Incremental feature selection based on rough set in dynamic incomplete data,” *Pattern Recognit.*, vol. 47, no. 12, pp. 3890–3906, 2014.

[163] F. H. Joaquín Derrac , Chris Cornelis , Salvador García, “Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection,” *Inf. Sci. (Ny).*, vol. 186, no. 1, pp. 73–92, 2011.

[164] S. W. Jue Wanga , Kun Guob, “Rough set and Tabu search based feature selection for credit scoring,” *Procedia Comput. Sci.*, vol. 1, no. 1, pp. 2425–2432, 2010.

[165] L. Ladha and T. Deepa, “Feature Selection Methods and Algorithms,” *Int. J. Comput. Sci. Eng.*, vol. 3, no. 5, pp. 1787–1797, 2011.

[166] L. A. S. Mark A. Hall, “Practical Feature Subset Selection for Machine Learning,” vol. Volume 20, p. 586, 1998.

[167] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution. Proceedings of the twentieth international conference on machine learning,” 2003.

[168] Y. Saeys, I. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[169] M. A. Hall, “Benchmarking Attribute Selection Techniques for Data Mining,” vol. 15, no. 6, pp. 1–15, 2000.

[170] I. Kononenko, “Estimating attributes: Analysis and extensions of RELIEF,” pp. 171–182, 1994.

[171] Yuhang Wang and F. Makedon, “Application of relief-F feature filtering algorithm to selecting informative genes for cancer classification using microarray data,” *Proceedings. 2004 IEEE Comput. Syst. Bioinforma. Conf. 2004. CSB 2004.*, no. Csb, pp. 477–478, 2004.

[172] M. A. Hall, “Correlation-based feature selection for discrete and numeric class machine learning,” 2000.

[173] A. G. Karegowda, A. S. Manjunath, G. Ratio, and C. F. Evaluation, “Comparative study of Attribute Selection Using Gain Ratio,” *Int. J. Inf. Technol. Knowl. Knowl. Manag.*, vol. 2, no. 2, pp. 271–277, 2010.

[174] G. H. J. YRon Kohavi, “Wrappers for feature subset selection Ron,” *Artif. Intell.*, no. 97, pp. 273–324, 1997.

[175] I. A. Gheyas and L. S. Smith, “Feature subset selection in large dimensionality domains,” *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.

[176] S. Maldonado and R. Weber, “A wrapper method for feature selection using Support Vector Machines,” *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2208–2217, 2009.

[177] M. Kudo and J. Sklansky, “Classifier-Independent Feature Selection for Two-stage Feature Selection,” *Adv. Pattern Recognit.*, vol. 1451, pp. 548–554, 1998.

[178] P. Bermejo, L. De La Ossa, J. A. Gámez, and J. M. Puerta, “Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking,” *Knowledge-Based Syst.*, vol. 25, no. 1, pp. 35–44, 2012.

[179] H. Liu, S. Member, L. Yu, and S. Member, “Algorithms for Classification and Clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, 2005.

[180] Y. Peng, Z. Wu, and J. Jiang, “A novel feature selection approach for biomedical data classification,” *J. Biomed. Inform.*, vol. 43, no. 1, pp. 15–23, 2010.

[181] M. A. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, and U. T. Nagar, “A Novel Feature Selection Approach for Intrusion Detection Data Classification,” *2014 IEEE 13th Int. Conf. Trust. Secur. Priv. Comput. Commun.*, pp. 82–89, 2014.

[182] N. Amor, S. Benferhat, and Z. Elouedi, “Naive bayes vs decision trees in intrusion detection systems,” *ACM Symp. Appl. Comput.*, pp. 420–424, 2004.

[183] M. Panda and M. R. Patra, “Network Intrusion Detection Using Naïve Bayes,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 258–263, 2007.

[184] F. Gharibian and A. A. Ghorbani, “Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection,” *Fifth Annu. Conf. Commun. Networks Serv. Res. (CNSR ’07)*, vol. 14, no. 3, pp. 350–358, 2007.

[185] S. Benferhat and K. Tabia, “On the combination of naive Bayes and decision trees for intrusion detection,” *Comput. Intell. Model. Control Autom. 2005 Int. Conf. Intell. Agents, Web Technol. Internet Commer. Int. Conf.*, vol. 1, pp. 211–216, 2005.

[186] J. L. Thames, R. Abler, and A. Saad, “Hybrid intelligent systems for network security,” *Proc. 44th Annu. southeast Reg. Conf. - ACM-SE 44*, p. 286, 2006.

[187] J. Zhang and M. Zulkernine, “Anomaly based network intrusion detection with unsupervised outlier detection,” *IEEE Int. Conf. Commun.*, vol. 5, no. c, pp. 2388–2393, 2006.

[188] J. Zhang, M. Zulkernine, and A. Haque, “Random-Forests-Based Network Intrusion,”

MAN Cybern., vol. 38, no. 5, pp. 649–659, 2008.

[189] and G. W. Iwan Syarif , Ed Zaluska, Adam Prugel-Bennett, "Application of Bagging, Boosting and Stacking to Intrusion Detection605," *Mach. Learn. Data Min. Pattern Recognition.Springer, Berlin, Heidelb.*, vol. 7376, pp. 539–602, 2012.

[190] D. P. Gaikwad and R. C. Thool, "Intrusion detection system using Bagging with Partial Decision Tree base classifier," *Procedia Comput. Sci.*, vol. 49, no. 1, pp. 92–98, 2015.

[191] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 80, 2004.

[192] A. Al-Shahib, R. Breitling, and D. Gilbert, "Feature selection and the class imbalance problem in predicting protein function from sequence," *Appl. Bioinformatics*, vol. 4, no. 3, pp. 195–203, 2005.

[193] T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 1, pp. 137–144, 2010.

[194] M. Wasikowski and X. W. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1388–1400, 2010.

[195] V. García, R. Alejo, J. S. Sánchez, J. M. Sotoca, and R. A. Mollineda, "Combined Effects of Class Imbalance and Class Overlap on Instance-Based Classification," *Intell. Data Eng. Autom. Learn. - Ideal 2006, Proc.*, vol. 4224, pp. 371–378, 2006.

[196] Srinivas Mukkamala and Andrew H. Sung, "Significant Feature Selection Using Computational Intelligent Techniques for Intrusion Detection," in *Advanced Information and Knowledge Processing*, no. January, 2005, pp. 293–314.

[197] S. B. Cho, "Incorporating soft computing techniques into a probabilistic intrusion detection system," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 32, no. 2, pp. 154–160, 2002.

[198] Y. Li, J. L. Wang, Z. H. Tian, T. B. Lu, and C. Young, "Building lightweight intrusion detection system using wrapper-based feature selection mechanisms," *Comput. Secur.*,

vol. 28, no. 6, pp. 466–475, 2009.

[199] Y. Chen, A. Abraham, and B. Yang, “Feature selection and classification using flexible neural tree,” *Neurocomputing*, vol. 70, no. 1–3, pp. 305–313, 2006.

[200] S. J. Horng *et al.*, “A novel intrusion detection system based on hierarchical clustering and support vector machines,” *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, 2011.

[201] Z. Zhou, S. Member, and X. Liu, “Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem,” vol. 18, no. 1, pp. 63–77, 2006.

[202] M. N. Mohammad, N. Sulaiman, and O. A. Muhsin, “A novel Intrusion Detection System by using intelligent data mining in WEKA environment,” *Procedia Comput. Sci.*, vol. 3, pp. 1237–1242, 2011.

[203] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, 2005.

[204] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” *Proc. 14th Int. Jt. Conf. Artif. Intell. - Vol. 2*, vol. 2, no. 0, pp. 1137–1143, 1995.

[205] R. E. Schapire, “Explaining adaboost,” *Empir. Inference Festschrift Honor Vladimir N. Vapnik*, pp. 37–52, 2013.

[206] C. W. Wang, “New ensemble machine learning method for classification and prediction on gene expression data,” *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, pp. 3478–3481, 2006.

[207] V. Engen, “Machine learning for network based intrusion detection: an investigation into discrepancies in findings with the KDD cup’99 data set and multi-objective evolution of neural network classifier ensembles from imbalanced data,” 2010.

[208] Gary M. Weiss, “Learning with Rare Cases and Small Disjuncts,” 1995, no. 1989, pp. 558–565.

[209] P. Chapman *et al.*, “CRISP-DM -Cross-Industry Standard Process for Data Mining-1.0 Step-by-step data mining guide.,” *Cris. Consort.*, p. 76, 2000.

[210] C. Shearer *et al.*, “The CRISP-DM model: The New Blueprint for Data Mining,” *J. Data Warehous.*, vol. 5, no. 4, pp. 13–22, 2000.