UNIVERSITY OF TECHNOLOGY, SYDNEY
AUSTRALIA

DOCTORAL THESIS

FEATURE SELECTION USING MUTUAL INFORMATION IN NETWORK INTRUSION DETECTION SYSTEM

Author:
Mohammed AMBUSAIDI

Supervisor:
Prof. Xiangjian He
Co-supervisor:
Dr. Priyadarsi Nanda
Co-supervisor:
A/Prof. Jinjun Chen

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

SCHOOL OF COMPUTING AND COMMUNICATIONS
THE FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

December 2015

Declaration of Authorship

I, Mohammed AMBUSAIDI, certify that the work in this thesis has not been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed: Production Note: Signature removed prior to publication.

Date: 3/12/2015

Abstract

FEATURE SELECTION USING MUTUAL INFORMATION IN NETWORK INTRUSION DETECTION SYSTEM

by Mohammed Ambusaidi

Network technologies have made significant progress in development, while the security issues alongside these technologies have not been well addressed. Current research on network security mainly focuses on developing preventative measures, such as security policies and secure communication protocols. Meanwhile, attempts have been made to protect computer systems and networks against malicious behaviours by deploying Intrusion Detection Systems (IDSs). The collaboration of IDSs and preventative measures can provide a safe and secure communication environment. Intrusion detection systems are now an essential complement to security project infrastructure of most organisations. However, current IDSs suffer from three significant issues that severely restrict their utility and performance. These issues are: a large number of false alarms, very high volume of network traffic and the classification problem when the class labels are not available.

In this thesis, these three issues are addressed and efficient intrusion detection systems are developed which are effective in detecting a wide variety of attacks and result in very few false alarms and low computational cost. The principal contribution is the efficient and effective use of mutual information, which offers a solid theoretical framework for quantifying the amount of information that two random variables share with each other. The goal of this thesis is to develop an IDS that is accurate in detecting attacks and fast enough to make real-time decisions.

First, a nonlinear correlation coefficient-based similarity measure to help extract both linear and nonlinear correlations between network traffic records is used. This measure is based on mutual information. The extracted information is used to

develop an IDS to detect malicious network behaviours. However, the current network traffic data, which consist of a great number of traffic patterns, create a serious challenge to IDSs. Therefore, to address this issue, two feature selection methods are proposed; filter-based feature selection and hybrid feature selection algorithms, added to our current IDS for supervised classification. These methods are used to select a subset of features from the original feature set and use the selected subset to build our IDS and enhance the detection performance.

The filter-based feature selection algorithm, named Flexible Mutual Information Feature Selection (FMIFS), uses the theoretical analyses of mutual information as evaluation criteria to measure the relevance between the input features and the output classes. To eliminate the redundancy among selected features, FMIFS introduces a new criterion to estimate the redundancy of the current selected features with respect to the previously selected subset of features.

The hybrid feature selection algorithm is a combination of filter and wrapper algorithms. The filter method searches for the best subset of features using mutual information as a measure of relevance between the input features and the output class. The wrapper method is used to further refine the selected subset from the previous phase and select the optimal subset of features that can produce better accuracy.

In addition to the supervised feature selection methods, the research is extended to unsupervised feature selection methods, and an Extended Laplacian score EL and a Modified Laplacian score ML methods are proposed which can select features in unsupervised scenarios. More specifically, each of EL and ML consists of two main phases. In the first phase, the Laplacian score algorithm is applied to rank the features by evaluating the power of locality preservation for each feature in the initial data. In the second phase, a new redundancy penalization technique uses mutual information to remove the redundancy among the selected features. The final output of these algorithms is then used to build the detection model.

The proposed IDSs are then tested on three publicly available datasets, the KDD Cup 99, NSL-KDD and Kyoto dataset. Experimental results confirm the effectiveness and feasibility of these proposed solutions in terms of detection accuracy, false alarm rate, computational complexity and the capability of utilising unlabelled data. The unsupervised feature selection methods have been further tested on five more well-known datasets from the UCI Machine Learning Repository. These newly added datasets are frequently used in literature to evaluate the performance of feature selection methods. Furthermore, these datasets have different sample sizes and various numbers of features, so they are a lot more challenging for comprehensively testing feature selection algorithms. The experimental results show that ML performs better than EL and four other state-of-art methods (including the Variance score algorithm and the Laplacian score algorithm) in terms of the classification accuracy.

Acknowledgements

I am pleasure to sincerely thank to my supervisor, **Professor Xiangjian He** 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. I owe my research achievements to his experienced supervision.

I would like to thanks my co-supervisor, **Dr. Priydarsi Nanda** for his friendly guidance, valuable suggestions and feedback. His encouragement and support have been a great help and kept me moving ahead at a critical time. I gratefully acknowledge the useful discussions with him. I would also like to thanks my co-supervisor, **A/Prof. Jinjun Chen** for his support and friendly advice which has been a great help.

I am extremely thankful to my fellow research colleagues and the staff of the school, especially those people listed below for providing various assistance for the completion of this research work.

- Professor Massimo Piccardi, Professor Doan B. Hoang, Associate Professor Qiang Wu, Dr. Min Xu, Dr. Wenjing Jia, Dr. Zhiyuan Tan, Dr. Aruna Jamdagni, Dr. Chao Zeng, Khaled Aldebei, Ahmed Mian Jan, Shaukat Abedi, Sari Awwad, Huiling Zhou, Sheng Wang, Minqi Li, Guopeng Zhang, Liangfu Lu and Wenbo Wang.

Special thanks to my wife, Intisar Alsabari, for her patience, understanding and assistance. I also thank my father Mr. Abdullah Ambusaidi and my mother Mrs. Azza Alharasi for the freedom to study for the long time necessary to complete postgraduate studies. I also would like to thank my sons, Awab Ambusaidi and Yassin Ambusaidi, for their patience. This thesis could not have been completed without the support and encouragement of my siblings. My special thanks go to my friends for their continuous support and encouragement. I would like to thank

Mr. John Hazelton for his English corrections. Last but not least, I would like to thank my sponsor, the Ministry of Higher Education, Oman, for providing me this opportunity to complete my PhD and for the fiancial assistance.

Contents

List of Figures

List of Tables

Abbreviations

ANN	**A**rtificial **N**eural **N**etworks
DoS	**D**enial **o**f **S**ervice
DR	**D**etection **R**ate
DT	**D**ecision **T**rees
E-LGFS	**E**xtended **L**ocal and **G**lobal structure preserving based on **F**eature **S**election
FE	**F**eature **E**xtraction
FMIFS	**F**lexible **M**utual **I**nformation **F**eature **S**election
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
FS	**F**eature **S**election
GSAD	**G**eometrical **S**tructure **A**nomaly **D**etection
HIDS	**H**ost based **I**ntrusion **D**etection **S**ystem
ICMP	**I**nter **C**ontrol **M**essage **P**rotocol
IDS	**I**ntrusion **D**etection **S**ystem
IFFS	**I**mproved **F**orward **F**loating **S**election
IT	**I**nformation **T**heory
LCC	**L**inear **C**orrelation **C**oefficient
LGFS	**L**ocal and **G**lobal structure preserving based on **F**eature **S**election
LSSVM	**L**east **S**quare **S**upport **V**ector **M**achine

MARS	**M**ultivariate **A**daptive **R**egression **S**plines
MCA	**M**ultivariate **C**orrelation **A**nalysis
MI	**M**utual **I**nformation
MIFS	**M**utual **I**nformation **F**eature **S**election
MIFS-U	**M**utual **I**nformation **F**eature **S**election-**U**niform information distribution
MMIFS	**M**odified **M**utual **I**nformation **F**eature **S**election
mRMR	**M**in-**R**edundancy **M**ax-**R**elevance
NCC	**N**onlinear **C**orrelation **C**oefficient
NIDS	**N**etwork based **I**ntrusion **D**etection **S**ystem
NI	**N**ormalised **M**utual **I**nformation
NLGFS	**N**ormalised **L**ocal and **G**lobal structure preserving based on **F**eature **S**election
NMIFS	**N**ormalised **M**utual **I**nformation **F**eature **S**election
PCA	**P**rincipal **C**omponent **A**nalysis
PCC	**P**earson's **C**orrelation **C**oefficient
PCC-R	**P**earson's **C**orrelation **C**oefficients-**R**ank
PDF	**P**robability **D**ensity **F**unction
R2U	**R**emote to **U**ser
RP	**R**edundancy **P**enalization
SVM	**S**upport **V**ector **M**achine
TANN	**T**riangle **A**rea based **N**earest **N**eighbours
TCP	**T**ransmission **C**ontrol **P**rotocol
TN	**T**rue **N**egative
TP	**T**rue **P**ositive

Publications from this Thesis

JOURNAL PUBLICATIONS:

[1] A. M. Ambusaidi, Z. Tan, X. He, P. Nanda, L. F. Lu, A. Jamdagni, Intrusion detection method based on nonlinear correlation measure, International Journal of Internet Protocol Technology 8 (2) (2014) 77-86.

[2] A. M. Ambusaidi, X. He, Z. Tan, P. Nanda, Building an intrusion detection system using a filter-based feature selection algorithm, IEEE Transactions on Computers, undergo a major revision.

[3] A. M. Ambusaidi, X. He, P. Nanda, Chen, J, Unsupervised Feature Selection Method for Machine Learning, Journal of Network and Computer Applications, to be submitted.

CONFERENCE PUBLICATIONS:

[1] A. M. Ambusaidi, L. F. Lu, X. He, Z. Tan, A. Jamdagni, P. Nanda, A nonlinear correlation measure for intrusion detection, in: International Conference on Frontier of Computer Science and Technology (FCST-12), IEEE, 2012, pp. 1-7.

[2] A. M. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, U. T. Nagar, A novel feature selection approach for intrusion detection data classification, International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-14), IEEE, 2014, pp. 82-89.

[3] A. M. Ambusaidi, X. He, P. Nanda, Unsupervised Feature Selection Method for Intrusion Detection System, International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-15), IEEE, 2015, Accepted.

Dedicated to My Family

Chapter 1

Introduction

This thesis addresses three important issues that affect the performance of anomaly intrusion detection systems. These issues are: a large number of false alarms, large-scale data in supervised learning applications and the absence of labels in an unsupervised data classification. Section 1.1 of this chapter outlines the background about intrusion detection systems. The motivations for the work presented in this thesis and objectives are discussed in Section 1.2. The contributions and novelty of the work are discussed in Section 1.3, followed by an outline of the structure of the remainder of the thesis in Section 1.4.

1.1 Background

Despite increasing awareness of network security, the existing solutions remain incapable of fully protecting internet applications and computer networks against the threats from ever-advancing cyber attack techniques. Developing effective and

adaptive security approaches, therefore, has become more critical than ever before. The traditional security techniques, as the first line of security defence, such as user authentication, firewall and data encryption, are insufficient to fully cover the entire landscape of network security while facing challenges from ever-evolving intrusion skills and techniques [2]. Hence, another line of security defence is highly recommended, such as IDS. Recently, an Intrusion Detection System (IDS) alongside anti-virus software has become an important complement to the security infrastructure of most organisations. The combination of these two lines of defence provides a high level of defence and strengthens network security against those threats.

Intrusion detection is the art of discovering and detecting network traffic or events on host machines that present anomalous behaviours or cause violations of regulations. It plays a significant role in monitoring and analysing daily activities occurring in computer systems to detect occurrences of security threats. However, infiltration techniques have become more sophisticated and have posed several challenges to the security tools. Thus, there is a need for an efficient and reliable IDS to safeguard computer networks from known as well as unknown attacks. To fulfil this purpose, an IDS is required to be accurate in discovering intruders and fast enough in order to make real-time decisions.

Intrusion detection techniques can be generally classified into two main categories. The first category is signature-based or misuse-based detection systems that detect on-going anomalies by looking for a match with any pre-defined attack signature [3]. These systems are widely used because they are simple and efficient. More importantly, they have a small number of false positive alarms. However, one of the disadvantages of these systems is that the detection accuracy and efficiency heavily depend on the quality of attack signatures. To extract such high quality

signatures requires the involvement of experts who have done extensive study of malicious behaviours, which is costly and time consuming. In addition, the signature of an intrusion is required before the system can detect the respective intrusion. Consequently, this type of IDS cannot detect any previously unknown attacks due to the lack of attack signatures. These limitations make systems or networks that have been protected by signature-based detection systems become vulnerable to those previously unknown attacks at any time. In addition, such weaknesses cause more critical issues in real practice because increasingly new and sophisticated infiltration techniques have been developed to defeat the security tools. Thus, the signature database of these systems needs to be continually updated in order to detect new attacks.

The second category is anomaly-based detection systems, which have been in favour with the research community. Anomaly-based detection makes an assumption that intruders' behaviours are different from those of normal network traffic [4, 5]. In comparison with signature-based detection systems, anomaly-based detection systems enjoy the advantage of detecting unknown attacks and variants of known attacks. That is because they make use of statistical analysis to evaluate the deviations of the behaviours of observed traffic flows from those of the normal traffic. They study normal traffic behaviours on a network and then create models for normal flows. After that, any deviations from the normal flows are considered as suspicious behaviours. The main advantages of these approaches are the ability of recognising known and unknown attacks, and there is no need for a continuous update of the attack knowledge base. However, the major weaknesses of these techniques include that they are prone to a large number of false alarms with newly occurring normal network traffic and poor detection efficiency with attacks that

mimic normal network traffic behaviours, and are not good at handling a large volume of data. In addition, the availability of labelled data for training the detection models is usually a major issue.

This thesis intends to address these limitations and focuses on developing anomaly-based detection systems which can be applied efficiently in detecting a wide variety of attacks. Although there is a current research direction to make use of the correlations in building intrusion detection systems, most of the proposed systems [1, 6–8] are based on linear correlation measures, such as Pearson's Correlation Coefficient, which are only capable of studying the linear correlations in a given sample set. However, the existence of nonlinear correlation hidden in a sample set limits the capability of these anomaly-based detection systems in extracting such correlation and therefore the systems are vulnerable to increasing number of attacks. In this thesis, a nonlinear correlation coefficient-based similarity measure is used to help extract both linear and nonlinear correlations between network traffic records. In addition, two supervised feature selection algorithms are proposed to cope with the issue of large-scale data. Furthermore, an unsupervised feature selection algorithm that can utilise unlabelled data is developed to select the best subset of features from the original dataset. This subset is then used to train the detection model.

1.2 Motivations and Objectives

Motivations

The quality of an anomaly detection system is defined by its effectiveness and adaptability. The effectiveness of a system is evaluated via its detection (true positive) rates as well as its false alarm (false positive) rates and the adaptability is measured by the ability of detecting known intrusions as well as new intrusions. However, three main challenges have to be overcome through the development of high quality anomaly based detection systems, which **motivate** this PhD research. These challenges are detailed as follows.

The first challenge is that the detection system must be capable of handling a large volume of data. These "big data" slow down the entire detection process and may lead to unsatisfactory classification accuracy due to the computational difficulties in handling such data. As a well-known intrusion evaluation dataset, KDD Cup 99 dataset is a typical example of large-scale datasets. This dataset consists of more than five millions of training samples and two millions of testing samples respectively. Such a large scale dataset may retard the building and testing processes of a classifier, or makes the classifier unable to perform due to system failures caused by insufficient memory.

The second challenge is the unavailability of class labels of training data. Classifying data in an unsupervised learning application, when the data labels are not available, is much more difficult than in supervised learning scenarios [9]. This is because in most of the real-world applications, the class labels are unknown which makes those intrusion detection systems based on supervised learning techniques not applicable.

The third challenge is that the detection system must accurately detect attacks with very few false alarms. It is very essential for intrusion detection systems to keep the number of false alarms as low as possible in order to maintain the level of security and reliability of networks. This technical challenge has been along with the development of IDSs since 1990s. A significant amount of work has been conducted attempting to address this issue. Numerous machine learning techniques, including Bayesian network [10], support vector machine [11] and Markov models [12], have been used to strengthen the detection capability of IDSs. However, the false alarm rate of those systems is still high.

Objectives

The overall aim of this thesis is to develop a novel anomaly-based IDS that is accurate in detecting attacks with low false alarms, able to handle large-scale data and fast enough to make real-time decisions. The specific objectives of this thesis are listed as follows.

1. We will propose an anomaly-based IDS based on mutual information measure, which is used to quantify the amount of information shared between two random variables.

2. We will propose a novel framework for supervised feature selection which considers the correlation among features. A filter-based supervised feature selection algorithms is developed in this thesis.

3. We will propose a supervised hybrid feature selection method to enhance the performance of the above filter method.

4. We will develop a novel framework for unsupervised feature selection that is capable of selecting the most optimal subset of features for unsupervised classification of network intrusions.

1.3 Thesis Contributions

Owing to the increasing storage capacity of computing systems, much more informative data can now be stored. However, it is an expensive or even less possible task to analyse all of these data due to the slow grow of computational capacity of computing systems in companion with the increase of data. The following are the main research contributions of this thesis.

1. In Chapter 3 of this thesis, we develop a new anomaly detection framework that effectively detects attacks by investigating the correlations among network traffic records [13, 14]. The proposed approach has the following properties:

    - It is capable of extracting both linear and nonlinear correlations between network traffic records, and
    - It does not require any update of the attack knowledge base.

2. In Chapter 4, we develop a new information theoretic criterion referred to as Flexible Mutual Information Feature Selection (FMIFS) to measure the relevance of each input feature with the output class [15]. The proposed algorithm has the following properties:

- It effectively selects relevant features based on mutual information for feature ranking,
- Introduces a new measure of redundancy to reduce the bias of mutual information in favour of multivalued attributes and keep the value of MI on the closed interval [0,1], and
- It does not require a user-defined parameter such as β for the selection processes of the candidate feature set as is needed in most of the state-of-the art methods.

3. To enhance the performance of the method proposed in Chapter 4, a new feature selection method, based on a hybrid filter/wrapper model, is developed in Chapter 5. The hybrid feature selection algorithm consists of two phases [16]. The upper phase conducts a preliminary search for an optimal subset of features using FMIFS, in which the mutual information between the input features and the output class serves as a determinant criterion. The selected set of features from the previous phase is further refined in the lower phase in a wrapper manner to retain a proper set of features with respect to the classification accuracy. The algorithm includes an additional search step together with the backtracking step named "replacing the weak feature". This step attempts to find if replacing weak features in the current selected feature set with new features can provide better performance.

4. The proposed feature selection algorithms in Chapter 4 and Chapter 5 are supervised feature selection methods. While the labelled data needed by supervised feature selection methods can be scarce, there is usually no shortage of unlabelled data. Hence, developing unsupervised feature selection methods,

which can utilise this data, attracts the attention of many researchers. Therefore, in Chapter 6, we extend our research attention to unsupervised feature selection algorithms [17, 18]. Specifically, the methods propose new Redundancy Penalization (RP) technique based on mutual information to eliminate the redundancy among selected features.

1.4 Thesis Structure

The remainder of this thesis is organised as follows. **Chapter 2** provides a thorough review of the related work. It commences with a brief introduction of anomaly detection systems. Then, a review of some dependency measures that have been successfully applied to anomaly detection systems is given. After that, the review introduces the principle of feature selection and some of the related methods based on feature selection. **Chapter 3** presents the first attempt to use the mutual information method to extract the correlation among the input samples. The method is able to extract both linear and nonlinear correlations between network traffic records. The extracted correlative information is then used to train the proposed IDS to detect anomalous behaviours in the network. **Chapter 4** presents a filter-based feature selection algorithm using mutual information, named FMIFS, to search for the most optimal subset of features. The aim is to use the theoretical analysis of mutual information as evaluation criterion to measure the relevance between the input features and the output classes. The selected subset of features is then used to train the proposed IDS. **Chapter 5** proposes an enhancement to the classification performance of FMIFS. It introduces a hybrid algorithm for feature selection based on the combined filter and wrapper methods feature selection. The

best set of candidate features is chosen, in a wrapper manner, from the top of the ranking list by looking for the best subset that produces the highest classification accuracy. The chosen subset is then used in building the proposed detection model. **Chapter 6** presents an unsupervised feature selection method for the classification problem when the class labels are unknown. The algorithm is named the modified Laplacian score, ML in short. Finally, **Chapter 7** summarises the contributions of this thesis and suggest avenues for future research.

Chapter 2

Related Work

In this chapter, background information is given to introduce the reader to the works that have been achieved in the following chapters of this thesis. The outline of this chapter is as follows. Section 2.1 introduces the problem of anomaly detection systems and reviews some of existing anomaly detection methods. Section 2.2 presents an overview of some popular dependence measures, correlation coefficient, mutual information and nonlinear correlation coefficient that have been used to improve the detection performance and reduce the false alarm rate of anomaly detection systems. Section 2.3 is devoted providing details of the principle of feature selection. In this section, the general problem of feature selection is introduced and the different categories of feature selection showing the advantages and disadvantages of each one are presented in some detail. The most related techniques for supervised feature selection based on mutual information method are presented in this section showing the strengths and weaknesses of each one. The unsupervised feature selection problem and some of the related state-of-the-art methods are then

discussed in Section 2.4. Section 2.5 describes three of the well known intrusion detection datasets: KDD Cup 99, NSL-KDD and Kyoto 2006+ dataset that have been widely used to evaluate the performance of intrusion detection systems. Finally, a summary to the chapter is given in Section 2.6.

2.1 Anomaly Detection System

The definition of anomaly detection refers to the ability of discovering patterns in data that do not match expected normal behaviour. These non-matching patterns are often named anomalies, intruders or outliers in the computer network domain. Detecting anomalies in network traffic has been researched in the research community from as early as the 19th century [19]. Over time, several anomaly detection techniques have been proposed in the domain of network security such as classification, information theoretic, statistical and clustering detection methods [5]. In general, anomaly detection systems operate in a two-phase fashion: a training phase, where the detection model is trained using the training dataset, and a testing phase, where the test data is passed through the trained model to check if it contains any attack. The key point is to build a model of legitimate activities using the normal data, named as the normal profile, and any deviation from the normal profile will be considered as an anomaly [20].

During the past decades, various anomaly detection methods have been proposed in literature. For example, the solid mathematical foundations of Support Vector Machine (SVM) has attracted the attention of many IDS researchers [21]. A review of the most commonly used techniques, made by Tsai et al. [22], indicates that

statistically SVM is one of the most popularly used methods in the last decade. SVM attempts to divide data into multiple classes by creating a hyperplane, which helps minimise the classification error and maximise the geometric margin. Eskin et al. [23] built adaptive probabilistic detection models, which adopted three machine-learning algorithms, including SVM, clustering method and K-neighbor, to detect anomalies within a noisy data. Their algorithm applied machine learning techniques to estimate the probability distributions of the mixture for detecting the anomalies. Hu et al. [24] proposed an anomaly detection system based on system call data using Robust SVM to model system behaviour and distinguish between each process in the system as normal or abnormal. Mukkamala et al. in [25] investigated the possibility of assembling various learning methods, including Artificial Neural Networks (ANN), SVMs and Multivariate Adaptive Regression Splines (MARS), to detect intrusions. They trained five different classifiers to distinguish between the normal traffic and four different types of attacks. They compared the performance of each of the learning methods with their model and found that an ensemble of ANNs, SVMs and MARS achieved the best performance in terms of classification accuracies for all the five classes. Peddabachigari et al. [26] proposed an intelligent system model (DT–SVM) based on a combination of Decision trees (DT) and support vector machines (SVM). Experiments on KDD Cup 99 dataset have shown that DT-SVM achieved high detection rate but has required a lot of computation time when dealing with big datasets. Recently, Chandrasekhar and Raghuveer [27] utilised neuro-fuzzy and radial SVM to build their detection approach. Their framework consists of four main steps: initial clustering, fuzzy neural network training, formation of SVM and classification using radial SVM. Toosi et al. [28] combined a

set of neuro-fuzzy classifiers in their design of the detection system, in which a genetic algorithm was applied to optimise the structures of neuro-fuzzy systems used in the classifiers. Based on the pre-determined fuzzy inference system (for example, classifiers), detection decision was made on the incoming traffic. The KDD Cup 1999 dataset was used to evaluate this neuro-fuzzy based detection system.

The aforementioned detection methods attempt to protect computer networks against intrusions by passing incoming traffic through the trained classifier. The performance of these systems could be further enhanced by using dependency measures to extract the correlation between samples that are then used to train the classifier of the detection system.

2.2 Dependency Measures for Anomaly Detection

Measuring the relevance between two random variables is an important and a fundamental problem. It has been used in many applications in several domains, such as statistics, economics and signal processing [29]. During the past decades, several researches have been conducted to develop a measure that can sensibly present the relevant relationship between two random variables. Correlation Coefficient and Mutual Information measures are the most popular dependence measures that have been widely applied in different domains. A brief overview of both measures is given below. In addition, an overview of the nonlinear correlation coefficient measure is also given below.

2.2.1 Correlation Coefficient

The correlation coefficient measure is one of the basic and most popular linear correlation methods used to measure linear dependence between two random variables [30]. It is commonly used in many areas due to its simplicity, low computational cost and ease of estimation. For any two random variables, their correlation coefficient indicates the magnitude of the relationship between the two variables and it is equal to the quotient of their covariance and the product of their standard deviations.

Given two random variables X and Y, as shown in Equation (2.1) and Equation (2.2) respectively,

$$X = \{x_1, x_2, \ldots, x_n\}, \tag{2.1}$$

$$Y = \{y_1, y_2, \ldots, y_n\}, \tag{2.2}$$

where n is the total number of samples. The correlation coefficient $p(X, Y)$ of the variables X and Y is defined as:

$$p(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{X})^2 \sum_{i=1}^{n}(y_i - \overline{Y})^2}}, \tag{2.3}$$

where $cov(X, Y)$ is the covariance between X and Y, and σ_X and σ_Y are the standard deviations and can be defined as:

$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y}), \tag{2.4}$$

$$\sigma_X = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{X})^2}, \tag{2.5}$$

$$\sigma_Y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{Y})^2}, \tag{2.6}$$

and $\overline{X}$ and $\overline{Y}$ indicate the means of X and Y, respectively and are defined as:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{2.7}$$

$$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{2.8}$$

The value of the correlation coefficient in Equation (2.3) is within the range [0,1]. It indicates the degree of the linear correlation between the two random variables. When the value of $p(X,Y)$ is close to 1 or -1, it denotes a strong relationship. If the value is close to 0, it means a weak relationship between the two variables. A positive correlation coefficient denotes that the two variables are in the same direction, and a negative one indicates that the two variables are in the opposite direction.

Many researchers consider the correlation among traffic samples to distinguish normal traffic from abnormal. Beauquier et al. in [31] for example, proposed a model

named Pearson's Correlation Coefficients-Rank (PCC-R), which applied PCC to evaluate distances between network traffic records. The experimental results have shown a slight enhancement in the false alarm rates compared to other comparative methods. Another attempt is done by Jin et al. in [6]. They utilised a covariance matrix of sequential samples to detect multiple network attacks. In order to investigate the performance of their model, they applied two different statistical pattern recognition approaches, namely threshold based detection approach and traditional decision tree approach, to detect anomalies. The experimental results have shown that both approaches can distinguish multiple known attacks in the covariance feature space effectively. However, one of the limitations of both models is susceptibility to any attacks which linearly change the monitored features.

Some new ideas were proposed in recent years based on different linear correlation techniques to deal with the problem of the linear changes of the monitored features and to reduce the false positive rate. In 2009, a method based on Triangle Area based Nearest Neighbours (TANN) was proposed by Tsai et al. in [7]. TANN combined clustering and classification techniques to detect attacks. Compared with the previously proposed methods, TANN shows a significant enhancement in the detection rate and false positive rates. In 2010, Jamdagni et al. proposed the Geometrical Structure Anomaly Detection (GSAD) model in [32]. GSAD is a pattern recognition method using the Mahalanobis Distance Map (MDM) to extract correlations between packet payload features. To reduce the processing overhead of the GSAD model, Tan et al. in [33] proposed a two-tier system based on the linear discriminant method. More recently, Tan et al. in [34] proposed an effective Multivariate Correlation Analysis (MCA) technique that investigates geometrical correlations (triangle areas) between features in a single network traffic record.

However, even though the linear correlation coefficient measure is widely used in different fields, it has some limitations that make it not always suitable to all applications. For example, it is well known that if two random variables are uncorrelated, in which their correlation coefficient is equal to zero, they are not necessarily independent of each other [30]. In addition, considering that in real world communication, the correlations can also be nonlinear. These limitations can increase the rate of false alarms of an IDS. To address these limitations, information theory provides a solid theoretical framework for analysing the information content of a data using various measures such as entropy, mutual information and others [35].

2.2.2 Mutual Information

Mutual Information (MI) has successfully addressed some limitations of the correlation coefficient. It provides a generalised correlation analogous to linear correlation coefficient, but it is sensitive to both linear and nonlinear correlations [36]. The key concept of mutual information is from information theory which was proposed in 1948 by Shannon [37]. It describes the amount of information shared between two random variables. It is a symmetric measure of the relationship between two random variables, and it yields a non-negative value [35]. A zero value of MI indicates that the two observed variables are statistically independent.

Given two continuous random variables $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$, where n is the total number of samples, the mutual information between X and Y is defined in Equation (2.9).

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{2.9}$$

where $H(X)$ and $H(Y)$ are the information entropies of X and Y. The information entropy is a measure of uncertainty of random variables X and Y, where $H(X)$ and $H(Y)$ are defined in Equation (2.10) and Equation (2.11) respectively.

$$H(X) = -\int_x p(x) \log p(x) dx, \tag{2.10}$$

$$H(Y) = -\int_y p(y) \log p(y) dy. \tag{2.11}$$

The *H(X, Y)* is the joint entropy of X and Y and is defined as

$$H(X,Y) = -\int_x \int_y p(x,y) \log p(x,y) dx dy. \tag{2.12}$$

Therefore, to quantify the amount of knowledge on variable X provided by variable Y (and vice versa), which is known as mutual information, Equation (2.13) is used.

$$I(X;Y) = \int_x \int_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dx dy, \tag{2.13}$$

where $p(x,y)$ is a joint probability density function (pdf), and $p(x)$ and $p(y)$ are the marginal density functions and are defined as

$$p(x) = \int p(x,y) dy \tag{2.14}$$

and

$$p(y) = \int p(x, y)dx. \tag{2.15}$$

For discrete variables, mutual information between two discrete random variables with a joint probability mass function $p(x,y)$ and marginal probabilities $p(x)$ and $p(y)$ is defined by replacing the integration notation with the summation notation as shown in Equation (2.16),

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{2.16}$$

and the entropies of X and Y are defined as:

$$H(X) = -\sum_{x \in X} p(x) \log p(x), \tag{2.17}$$

$$H(Y) = -\sum_{y \in Y} p(y) \log p(y). \tag{2.18}$$

Several anomaly detection methods based on mutual information have been proposed in the literature. Das and Schneider in [38, 39] applied mutual information to extract the dependence between all attribute sets and proposed an anomaly detection system which attempts to find unexpected behaviour and flag it as an anomaly attempt. They set a threshold of mutual information and obtain a set of dependent attribute pairs. Based on the results in this set, an anomaly factor for each individual sample is defined. Kopylova et al. [40] investigated the use of mutual information in network traffic anomaly detection using generalized Renyi entropy

rather than the traditional Shannon entropy measure. The generalized Renyi entropy measures the uncertainty and complexity of a collection of data samples. For a system X, where $X = \{x_1, x_2, \ldots, x_m\}$ for a finite set of m possible states, the Shannon entropy of X is given by Equation (2.19)

$$H(X) = -\sum_{i=1}^{m} P(x_i) log P(x_i), \tag{2.19}$$

where $P(x_i)$ indicates the probability of the system being in state x_i.

Since the value of mutual information does not range in a closed interval [0,1] as the correlation coefficient does, to indicate the degree of the nonlinear correlation, with 0 and 1 denotes the weakest and the strongest relation, respectively. To address this, the proposed anomaly detection systems in Chapter 3 of this thesis applies a revised version of the mutual information measure [41] to extract both linear and nonlinear correlations between network traffic records, where the value of the revised version ranges in [0,1], and uses the extracted information to detect anomalies. Chapter 3 of this thesis is developed based on the works published in [13, 14].

In general, an IDS deals with a large volume of data that consists of a great number of traffic patterns. Each pattern in a dataset is characterised by a set of features (or attributes) that are represented as a point in a multi-dimensional feature space. A pattern may contain irrelevant and redundant features that slow down the training and testing processes or even affect the classification performance by causing more mathematical complexity. In practice, however, it is worthwhile to keep the number of features as small as possible in order to reduce the computational cost and the complexity of building a classifier. Therefore, the performance of the aforementioned systems could be further improved by introducing an additional step, dimensionality

reduction, as part of the preprocessing stage to eliminate these unimportant features from the used dataset. Dimensionality reduction, such as feature extraction and feature selection, has been successfully applied to machine learning and data mining to solve this problem. Feature Extraction (FE) techniques attempt to transfer the input features into a new feature set, while Feature Selection (FS) algorithms search for the most informative features from the original input data [42]. This research focuses on feature selection.

2.2.3 Nonlinear Correlation Coefficient

The disadvantage of MI is that it does not range in a definite closed interval [0, 1] as the correlation coefficient does [41, 43, 44]. Therefore, Wang et al. [41] proposed a revised version of the MI, named Nonlinear Correlation Coefficient, NCC in short.

To explain the Nonlinear Correlation Coefficient, we refer to the definitions proposed in [41, 43, 44]. Considering two random variables $X = \{x_i\}_{1 \leq i \leq N}$ and $Y = \{y_i\}_{1 \leq i \leq N}$. Their values are first sorted in ascending order and placed into b ranks with the first N/b values in the first rank, the second N/b values into the second rank, and so on. Second, the sample pairs, $\{(x_i, y_i)\}_{1 \leq i \leq N}$, are placed into a $b \times b$ rank grids by comparing the sample pairs to the rank sequences of X and Y.

After the processing in such a manner, the probability of a variable for state i is $p_i = \frac{N/b}{N} = \frac{1}{b}$, and the joint entropy of the two variables is $p_{ij} = \frac{n_{ij}}{N}$, where n_{ij} is the number of samples pairs distributed into the ij^{th} rank grid. The Nonlinear Correlation Coefficient (NCC) is defined in Equation (2.20)

$$NCC(X;Y) = H^r(X) + H^r(Y) - H^r(X,Y), \tag{2.20}$$

where $H^r(X)$ is the revised entropy of the variable X, which is defined as.

$$H^r(X) = -\sum_{i=1}^{b} p_i log_b p_i \tag{2.21}$$

And $H^r(X,Y)$ is the revised joint entropy of the two variables X and Y, which is given by Equation (2.22)

$$H^r(X,Y) = -\sum_{i=1}^{b}\sum_{j=1}^{b} p_{ij} log_b p_{ij} = -\sum_{i=1}^{b}\sum_{j=1}^{b} \frac{n_{ij}}{N} log_b \frac{n_{ij}}{N}, \tag{2.22}$$

Considering that the probability p_i, for every state of variable X or Y, is constantly $\frac{1}{b}$, and the way in which the N value pairs are distributed into the $b \times b$ rank grids indicates the statistical general relation between the two variables. Furthermore, the number of samples distributed into each rank of X and Y is fixed, so

$$H^r(X) = -\sum_{i=1}^{b} \frac{\frac{N}{b}}{N} log_b \frac{\frac{N}{b}}{N} = -b * \frac{1}{b} log_b \frac{1}{b} = 1. \tag{2.23}$$

Similarly, we have that $H^r(Y) = 1$. Moreover, the nonlinear correlation coefficient can be rewritten as

$$NCC(X;Y) = 2 * (-\sum_{i=1}^{b} \frac{1}{b} log_b \frac{1}{b}) + \sum_{i=1}^{b}\sum_{j=1}^{b} p_{ij} \log_b \ p_{ij}. \tag{2.24}$$

Therefore,

$$NCC(X;Y) = 2 + \sum_{i=1}^{b}\sum_{j=1}^{b} p_{ij} \log_b \ p_{ij} = 2 + \sum_{i=1}^{b}\sum_{j=1}^{b} \frac{n_{ij}}{N} log_b \frac{n_{ij}}{N}. \tag{2.25}$$

NCC is sensitive to the nonlinear type of correlation between two variables. It can describe this type of relationship with a number in a closed interval [0,1], where 0 indicates the minimum general correlation and 1 indicates the maximum one. If the sample sequences are exactly the same, the last term in Equation (2.25) equals to -1 and thus, $NCC(X;Y) = 1$. On the other hand, if the two variables are completely uncorrelated, the sample pairs distribute equally into the $b \times b$ ranks, the sum equals to -2 and thus, $NCC(X;Y) = 0$.

2.3 Feature Selection Based on Mutual Information

Feature selection is a technique for eliminating irrelevant and redundant features and selecting the most optimal subset of features that produce a better characterisation of patterns belonging to different classes. The feature selection problem has been around since the early 1970's. Due to its computational complexity, it still remains an open problem for researchers. Feature selection reduces computational cost, facilitates data understanding, improves the performance of modelling and prediction and speeds up the detection process [45].

A feature f_i in a feature space is relevant to the class if it embodies useful information about the class and its removal degrades the performance of the classification.

The irrelevant feature is the one that does not contain any useful information about the class and its existence degrades the performance of the classification [46]. An irrelevant feature can be a redundant feature or a noisy feature. The redundant feature cannot provide any additional information to the classification after selecting the S subset of features because another feature has already given the same information. The noisy feature, which is not redundant does not contain any information about the class.

Methods for feature selection are generally classified into three main categories: *filter*, *wrapper* and *hybrid* approaches. Filter algorithms start searching from an empty subset and utilise an independent measure (such as, information measures, distance measures, or consistency measures) as a criterion to estimate the relation of a set of features. The searching process continues until a predefined stopping criterion (for example, the search is completed, a desired number is reached, or adding or deleting of any feature does not produce a better feature subset) is met. An optimal subset of features that can provide the best representation of the data is output from the algorithm. This approach is argued to be less computationally expensive, easily applied to high-dimensional datasets and more general. However, their results are not always acceptable. Due to the lack of interaction between the classifier and the dependence among features, filter methods might fail to choose the best available subset or might select redundant features [47]. Thus, the classification performance of the learning models built based on these selected features is varied and highly dependent on the quality of the selection criterion.

Wrapper algorithms utilise a particular learning algorithm (such as, the decision tree or SVM) as a fitness function to evaluate the goodness of features. The searching

process for an optimal subset of features continues until a predefined stopping criterion is achieved. In comparison with filter methods, wrapper methods are argued to be more accurate. However, wrapper approaches are often much more computationally complicated when dealing with high-dimensional data or large-scale data than filter approaches [48].

To cope with the aforementioned drawbacks and to avoid the burden of specifying a stopping criterion, many researchers attempt to exploit the advantages of both filter and wrapper methods. Hybrid algorithms utilise both an independent measure and a fitness evaluation function of the feature subset. They use the knowledge delivered by a filter algorithm and a specific machine learning algorithm to choose the final best subset of the feature [49]. As it has been claimed in [47], methods belonging to this category are not fast as the filter approaches, but are argued to be more effective and can achieve better classification performance.

In accordance with the existence of label of data or not, feature selection techniques are generally classified into three groups: supervised, semi-supervised and unsupervised feature selection. Supervised and semi-supervised methods are usually applied on labelled data, while the unsupervised method is more appropriate for unlabelled data [50]. In this work the focus will be more on supervised and unsupervised feature selection.

2.3.1 Supervised Feature Selection

Given a training dataset $T = D(F, C)$ with n features and m instances, where $F = \{f_1, ..., f_n\}$ and $D = \{i_1, ..., i_m\}$ are the sets of features and instances, respectively,

$C = \{c_1, ..., c_l\}$ represents the set of classes (or labels) which instances belong to. Let $J(S)$ be a criterion function of selecting a subset of features S from F, where $S \subseteq F$. Without any loss of generality, it can be assumed that a subset of features that achieve a higher value of $J(\cdot)$ demonstrates a better feature space. The task of supervised feature selection is formally defined as selecting a subset of features S from the original feature space F such that $J(S)$ is as high as possible.

2.3.2 Existing Supervised Feature Selection Methods

As stated above, a feature is relevant to the class if it contains important information about the class; otherwise it is irrelevant or redundant. Since mutual information is good at quantifying the amount of information shared by two random variables, it is often used as an evaluation criterion to evaluate the relevance between features and the class labels. Under this context, features with high predictive power are the ones that have larger mutual information $I(C; f)$. On the contrary, in the case of $I(C; f)$ equal to zero, the feature f and the Class C are proven to be independent from each other. This mean that feature f will contribute redundancy to the classification.

However, due to the reason that the value of the MI between attributes is used as a criterion to select features from the original set, any computational errors could result in a significant degradation of the accuracy of any feature selection algorithms based on this measure. Therefore, the computation of MI, which requires the estimation of probability density functions (pdfs) or entropies from the input data instances, is not an easy task. Thus, several estimation techniques could be applied to compute MI. Histogram and kernel density estimations are the most popular estimation methods to estimate the pdfs [51, 52]. Peng et al. [53] claimed that

the histogram approach was computationally efficient, but could produce a large number of estimation errors. They also stated that kernel density estimation had a high estimation quality and at the same time high computational load. Another significant challenge with histogram techniques is the restriction to a low-dimensional data space [54]. It has also been pointed out by Rossi [55] that both histogram and kernel density approaches suffer from the well-known curse of high-dimensionality. As this study is working with high-dimensional data, these two estimations are inapplicable.

To avoid the aforementioned problems, in this work, the estimator proposed by Kraskov et al. [56] is applied. Unlike histogram and kernel density estimations, this technique relies on estimating entropies from the data using an average distance of the k-nearest neighbour. The novelty of this estimator is its ability to estimate MI between two random variables of any data space. The main idea is to estimate the entropy, with or without knowing the densities $p(u,v)$, $p(u)$ and $p(v)$, based on the k-nearest neighbours algorithm. More details about estimating MI can be found in [56] and Appendix B.

Recently, mutual information has been used by a number of researchers to develop an information theoretic feature selection criteria [42, 53, 57–59]. Battiti [57] defined feature reduction as a process of selecting a subset S of the most relevant features from the original feature set F and proposed a feature selection algorithm, MIFS in short. Battiti's MIFS [57] harnessed MI between inputs and outputs for a single selection of features by calculating the $I(C; f_i)$ and $I(f_s, f_i)$, where f_s and f_i are candidate features and C is the class label. MIFS selects the feature that maximises $I(C; f_i)$, which is the amount of information that feature f_i carries about the class

C, and is corrected by subtracting a quantity proportional to the MI with the features selected previously.

Given an initial set F with n features, the task is to search for an optimal subset S that can produce the best classification accuracy, where $S \subset F$. MIFS is a heuristic incremental search algorithm and the selection process continues until a desired number of K inputs are selected. Equation (2.26) shows the criterion function of MIFS.

$$J_{MIFS} = I(C; f_i) - \beta \sum_{f_s \in S} I(f_i; f_s), \tag{2.26}$$

where β is a user-defined parameter that is applied to account for the redundancy between the candidate feature and the set of selected features.

As can be seen, Equation (2.26) consists of two terms. The left-hand side term, $I(C; f_i)$, represents the amount of information that feature f_i carries about the class C. A relevant feature is the one that maximises this term. The right-hand side term, $\beta \sum I(f_s; f_i)$, is used to eliminate the redundancy among the selected features.

In the follow-up research, various methods have been proposed to enhance Battiti's MIFS. Most of the studies have been conducted on the right-hand side term of Equation (2.26). Kwak and Choi in [58] made a better estimation of MI between input features and output classes and proposed a greedy selection algorithm named MIFS-U, in which U stands for uniform information distribution. The algorithm of MIFS-U differs from that of MIFS in the right-hand side term as shown in Equation (2.27).

$$J_{MIFS-U} = I(C; f_i) - \beta \sum_{f_s \in S} \frac{I(C; f_s)}{H(f_s)} I(f_i; f_s) \tag{2.27}$$

Despite the redundancy parameter β used in the aforementioned methods to help to control the redundancy among features, it remains an open question on how to choose the most appropriate values for these parameters. If the chosen value is too small, the redundancy between input features is not taken into consideration and therefore both relevant and redundant features are involved in the selection processes. If the chosen value is too large, the algorithms only consider the relation between input features rather than the relation between each input feature and the class [42]. Thus, it is hard to determine the value of the parameter. In addition, both MIFS and MIFS-U neglect the influence of the number of selected features. This reduces the influence of $I(C; f_i)$ on Battiti's MIFS and Kwak's MIFS-U when the term on the right-hand side in MIFS and MIFS-U increases, which is because this term is a cumulative sum [59]. This results in the irrelevant features being selected into the set S.

The min-Redundancy Max-Relevance (mRMR) [53] and Modified MIFS (MMIFS) [1] both show another variant of Battiti's MIFS criterion. The mRMR removes the burden of setting an optimal value for β and replaced it with $1/ \mid S \mid$. mRMR is defined in Equation (2.28)

$$J_{mRMR} = I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} I(f_i; f_s) \tag{2.28}$$

while MMIFS set the value of parameter β to be equal to $\beta^{'}/ \mid S \mid$, where $\beta^{'}$ is the redundancy parameter, as shown in Equation (2.29).

$$J_{MMIFS} = I(C; f_i) - (\frac{\beta'}{\mid S \mid}) \sum_{f_s \in S} I(f_i; f_s), \tag{2.29}$$

where $|S|$ in Equation (2.28) and Equation (2.29) is the cardinality of the set S, which is used to control the influence of the number of selected features since the right-hand side of the algorithm is a cumulative sum.

However, in the case of $\beta = 1/ \mid S \mid$ or $\beta = \beta'/ \mid S \mid$, in Equation (2.28) and Equation (2.29) respectively, then mRMR and MMIFS are equal to Battiti's MIFS. Therefore, the unbalance between the left and right hand sides in Equation (2.28) and Equation (2.29) remains unsolved totally in mRMR [59] and MMIFS [1]. This might result in selecting irrelevant features. In addition, similar to Battiti's MIFS and Kwak's MIFS-U, selecting an appropriate value for the parameter β' in MMIFS remains an open question.

Normalised Mutual Information Feature Selection (NMIFS) [59] is an improved version of mRMR. NMIFS introduced a better solution for the unbalance between the two terms in Equation (2.28) and Equation (2.29). The authors explained that in order to achieve a good balance between the two terms, the right-hand side of the equation should be normalised by the entropy of the current feature f_i and the selected feature f_s. Equation (2.30) shows the selection criterion of the NMIFS.

$$J_{NMIFS} = I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} NI(f_i; f_s) \tag{2.30}$$

where

$$NI(f_i; f_s) = \frac{I(f_i; f_s)}{min\{H(f_i), H(f_s)\}} \tag{2.31}$$

Vinh et al. in [60] stated that NMIFS solved the unbalance limitation only in the case of two classes and it might face a problem when the number of classes increases. That is because when dealing with multi-class problems the left-hand side of Equation (2.30) breaks the maximum bound 1, while the right-hand side takes values in the range [0,1]. This might lead to neglecting the value of the right term and therefore selecting noisy features. Vinh et al. in [60] proposed another modification to MIFS as shown in Equation (2.32).

$$f(X_i) = \frac{I(X_i; C)}{min\{H(X_i), H(C)\}} - \frac{1}{|S|} \sum_{X_s \in S} \frac{I(X_s; X_i)}{min\{H(X_s), H(X_i)\}}. \tag{2.32}$$

However, in the case of $I(X_s; X_i) \geq I(X_i; C)$, the value of Equation (2.32) may well be outside the closed interval [0,1] and this may lead to select irrelevant features.

Therefore, after investigating the aforementioned limitations of the related feature selection methods based on mutual information in the work [15] submitted to IEEE Transaction on Computers and [16], a new feature selection method for anomaly detection system is proposed. Chapter 4 and Chapter 5 of this thesis are developed based on the works in [15] and [16] respectively.

All of the feature selection algorithms discussed above and the proposed feature selection algorithms in Chapter 4 and Chapter 5 are supervised feature selection methods. These methods require labelled data. However, labelled data are not always available and are also hard or expensive to obtain which makes these methods

not able to be applied to such data. Hence, developing unsupervised feature selection algorithms, which can utilise unlabelled data, attracts the attention of many researchers. In Chapter 6 of this thesis, this research is extended to unsupervised feature selection method.

2.4 Unsupervised Feature Selection Methods

As mentioned in the above section, feature selection methods can be supervised, semi-supervised and unsupervised in regard to the availability of the class labels. However, many applications (such as real-world applications) do not contain any label, hence, the unsupervised feature selection process become difficult and hard to achieve [61]. This section discusses the related unsupervised feature selection methods to the work proposed in Chapter 6 of this thesis.

2.4.1 Unsupervised Feature Selection

Given a training dataset $D = \{x_i = [f_1, ..., f_n]\} \subset \Re^n$ without labels, with n features and m instances, $F = \{f_1, ..., f_n\}$ and $D = \{i_1, ..., i_m\}$ are the sets of features and instances, respectively. x_i represents the i-th data instances containing n-th features. Let $J(S)$ be a criterion function for selecting subset of features S from F. The task of unsupervised feature selection methods is to select an optimal feature subset S from the original feature space F where $J(S)$ is as high as possible. The basic assumption is that samples belonging to the same class are probably close to each other, otherwise they are from a different class.

2.4.2 Existing Unsupervised Feature Selection Methods

Selecting features in unsupervised learning applications is much harder than in the case of supervised learning where the class label is available. It is not an easy task to assess the relevance of a feature or a subset of features when there are no labels available with the data. Therefore, several attempts have been conducted to develop an unsupervised feature selection technique that can utilise this data. The Laplacian score [62] is one of the popular unsupervised feature selection methods that uses a k-nearest neighbour graph to investigate the locality preserving power of every features in the data.

Let L_r denote the Laplacian score of the $r-$th feature and f_{ri} denote the $r-$th feature of the $i-$th sample. Given a data $X = \{x_1, x_2, \ldots, x_n\}$, if x_i and x_j are close to each other, an edge with weight S_{ij} between both samples is built. This means that x_i is one of the k nearest neighbours of x_j or x_j is one of the k nearest neighbours of x_i. The weight matrix S_{ij} between sample x_i and x_j can be defined as:

$$S_{ij} = \begin{cases} exp(-\frac{d(x_i;x_j)^2}{t}), & if\ x_i \in kNN(x_j)\ or\ x_j \in kNN(x_i) \\ 0, & \text{otherwise} \end{cases} \tag{2.33}$$

where t is a suitable constant. A good feature is the one that minimises the following object function:

$$L_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{Var(f_r)}, \tag{2.34}$$

where, $Var(f_r)$ is the variance of the $r-$th feature. Equation (2.34) aims to select features that hold the best locality preserving power among the input features. Features that have large variance values are preferred. These features are expected to have the most representative power among others. Equation (2.34) can be further explained as follows.

For the matrix S, define the diagonal matrix $D_{ii} = \sum_j S_{ij}$ and the graph Laplacian matrix $L = D - S$. Based on D, the weighted data variance can be calculated using Equation (2.35).

$$Var(f_r) = \check{f}_r^T D \check{f}_r, \tag{2.35}$$

where

$$\check{f}_r = f_r - \frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1}, \tag{2.36}$$

A good feature is the one that has bigger S_{ij} and smaller $(f_{ri} - f_{rj})$. Thus,

$$\sum_{ij}(f_{ri} - f_{rj})^2 S_{ij} = 2f_r^T L f_r = 2\check{f}_r^T L \check{f}_r. \tag{2.37}$$

Therefore, the Laplacian score L_r of the r-th feature is given by Equation (2.38).

$$L_r = \frac{\check{f}_r^T L \check{f}_r}{\check{f}_r^T D \check{f}_r} \tag{2.38}$$

Unlike the Laplacian score, Local and Global structure preserving (LGFS) [9] not only considers the locality structure preserving power of each feature but also its globality structure preserving. The assumption behind LGFS methods is that samples belonging to the same class are probably located close to each other, otherwise far away from each other.

LGFS first extracts a k_1 nearest neighbours graph on X that has a similarity matrix S^n, where each sample x_i is linked with its k_1NN, as follows.

$$S^n_{ij} = \begin{cases} exp(-\frac{d(x_i;x_j)^2}{t}), & if\ x_i \in N_{k_1}(x_j)\ or\ x_j \in N_{k_1}(x_i) \\ 0, & \text{otherwise} \end{cases}$$

where $d(x_i, x_j)$ represents the Euclidean distance between x_i and x_j, and t is a suitable positive constant.

Second, LGFS constructs a k_2 farthest neighbourhood graph on X with similarity matrix S^f, where each sample x_i is linked with its K_2FN, as follows.

$$S^f_{ij} = \begin{cases} exp(-\frac{d(x_i;x_j)^2}{t}), & if\ x_i \in F_{k_2}(x_j)\ or\ x_j \in F_{k_2}(x_i) \\ 0, & \text{otherwise.} \end{cases}$$

Figure 2.1 shows the k_1 Nearest neighbourhood graph and F_{k_2} Farthest neighbourhood graph. From the figure, it can be seen that x_i and its k_1 nearest neighbours are connected to each other and belong to the same class, while x_i and its k_2 farthest neighbours belong to the different class.

After that, Dis matrix, where $Dis_{ij} = d(x_i; x_j)^2$ and $i, j = 1, 2, \ldots, m$, is calculated to select the features that have the best locality and globality preserving power

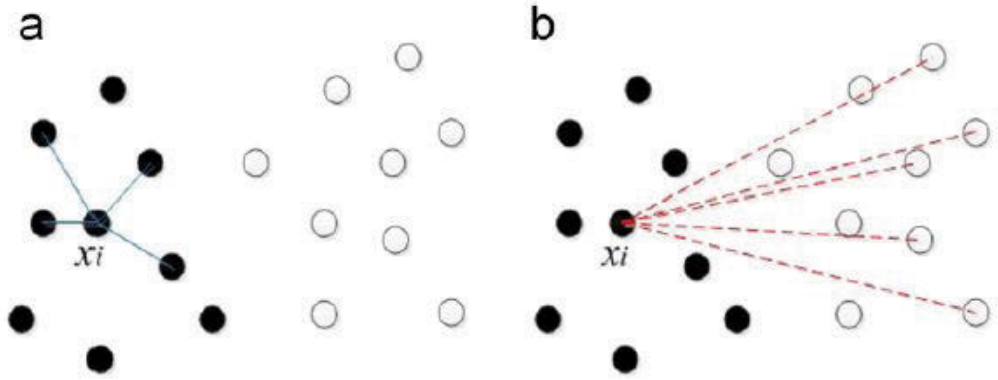

FIGURE 2.1: Nearest and farthest neighbourhood graph. (a) N_{k_1} Nearest neighbourhood graph, (b) F_{k_2} Farthest neighbourhood graph [9]

using Equation (2.39). The parameter $t > 0$ is set to be the mean value of all the elements in the matrix Dis. Then, the LGFS can be defined as follows.

$$LGFS = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}^f}{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}^n} \tag{2.39}$$

From Equation (2.39), it is clear that when two samples have a near edge, a good feature should have similar values on both of these two samples. On the other hand, when two samples have a far edge, a good feature should have large different values on both of these two samples and thus $LGFS$ should be maximised.

By defining the diagonal matrix D^n and L^n for S^n, where $D_{ii}^n = \sum_j S_{ij}^n$ and $L^n = D^n - S^n$ and similarly for S^f defining the diagonal matrix D^f and L^f, we get:

$$S^n = \sum_{ij}(f_{ri} - f_{rj})^2 S^n_{ij} = \sum_{ij}(f^2_{ri} + f^2_{rj} - 2f_{ri}f_{rj})S^n_{ij}$$
$$= 2\sum_{ij} f^2_{ri}S^n_{ij} - 2\sum_{ij} f_{ri}S^n_{ij}f_{rj}$$
$$= 2f_r^T D^n f_r - 2f_r^T S^n f_r = 2f_r^T L^n f_r$$

and

$$S^f = \sum_{ij}(f_{ri} - f_{rj})^2 S^f_{ij} = 2f_r^T L^f f_r.$$

Therefore, Equation (2.39) can be rewritten as follows.

$$LGFS_r = \frac{f_r^T L^f f_r}{f_r^T L^n f_r}, \tag{2.40}$$

However, as stated in [9], both Laplacian score and LGFS are not considering the redundancy among features, which might lead to the selection of redundant features and therefore affect the classification performance. Ren et al. in [9] proposed an Extended version of LGFS, named Extended Local and Global structure preserving (E-LGFS). E-LGFS applies the normalised mutual information method, that has been proposed in [59], to eliminate redundancy among selected features.

In E-LGFS, LGFS in Equation (2.40) is first normalised using a linear transformation to fix the value of LGFS to the range of $[0, 1]$, as shown in Equation (2.41).

$$NLGFS_r = \frac{LGFS_r - a}{b - a} \tag{2.41}$$

where a and b are the minimum and maximum of $\{LGFS_1, LGFS_2, ..., LGFS_n\}$, respectively. The value of $NLGFS_r$ is within the range of $[0, 1]$, where the value 0 or 1 indicates that $LGFS_r$ is the minimum or maximum of $\{LGFS_1, LGFS_2, ..., LGFS_n\}$ respectively.

Finally, the normalised mutual information method, which is defined in [59], is used to eliminate the redundancy among selected features using Equation (2.42).

$$NI(f_i; f_j) = \frac{I(f_i; f_j)}{min\{H(f_i), H(f_j)\}} \tag{2.42}$$

Thus, the final selection criterion of E-LGFS can be rewritten as follows:

$$J_i = NLGFS_i - NI(f_i; S) \tag{2.43}$$

However, E-LGFS has high computational complexity when dealing with high dimensional data and large-scale data.

Therefore, an Extended Laplacian score EL and a Modified Laplacian score ML methods are proposed in [17, 18]. These methods provide a solution for eliminating redundancies among selected features without the need for extracting the global structure information. Chapter 6 is developed based on the work in [17, 18].

2.5 Description of the Benchmark Datasets for Intrusion Detection

Currently, there are only a few public datasets for intrusion detection evaluation. According to the review by Tsai et al. [22], the majority of the IDS experiments are performed on the KDD Cup 99 datasets. Other two intrusion detection datasets are named as NSL-KDD dataset and Kyoto 2006+ dataset. Therefore, in order to facilitate a fair and rational comparison with other state-of-the-art detection approaches, we have selected these three datasets to evaluate the performance of our detection system.

2.5.1 KDD Cup 99 Dataset

KDD Cup 99 datasets is the most comprehensive dataset that is still widely applied to compare and measure the performance of IDSs [63–65]. This dataset was derived from the DARPA 1998 dataset. It contains training data, "10% KDD Cup 99", with approximately five millions of connection records and test data, "kddcup test-data", with about two millions of connection records. In addition, the KDD Cup 99 contains one more dataset named "Corrected labels KDD". Each record in these datasets is labelled as either normal or an attack, and it has 41 different quantitative and qualitative features. The 41 features are generally categorised into three main groups. The first group is the basic features (that is, attributes 1 to 9) that can be extracted from a TCP/IP connection. The second group refers to features 10 to 22 that are named as content-based features presenting the information derived from network packet payloads. The third group corresponds to the traffic-based features,

which are carried by features 23 to 41 of each record. A complete list of the set of features and the detailed description are given by Table 2.1.

The attacks are divided into four different types, namely Probe, Denial of Service (DoS), User to Root (U2R) and Remote to User (R2U). Table 2.2 shows a brief distribution of each attack.

The corrected labels KDD Cup 99 dataset has been used to validate some of state-of-the-art IDSs such as [28, 66–70]. Therefore, to make a fair comparison with those systems, we use this dataset to test the performance of our detection model. This set contains approximately 311,029 TCP/IP connection records, where around 74.4% of the samples are DoS attacks, and the remaining ones are distributed as follows: 19.4% normal, 1.33% probe, 4.73% R2L and 0.028% U2R traffic.

2.5.2 NSL-KDD Dataset

Even though KDD Cup 99 dataset is a well-known dataset and widely used for network-based intrusion detection techniques, it contains some problems such as including a huge number of redundant records, which affect the effectiveness of evaluated systems greatly as a consequence. To overcome these issues, Tavallaee et al. [71] in 2009, presented a new revised version of KDD Cup 99 named as NSL-KDD. The KDDTrain$^+$ and KDDTest$^+$ sets of NSL-KDD dataset consist of approximately 125,973 and 22,544 connection records respectively. Similar to KDD Cup 99, each record in this data is unique with 41 features and labelled as normal or attack. NSL-KDD dataset contains the same four types of attacks as the original KDD 99 dataset.

TABLE 2.1: The different group of features in KDD Cup 99 and NSL-KDD dataset

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

TABLE 2.2: The different types of attacks and description

Attack type	Attack name	Description
Probing	nmap ipsweep portsweep satan	The attacker in Probe attacks scans a network searching for important information about target computers.
DoS	back land neptune pod smurf teardrop	The attacker in DoS attacks sends many requests to network resources to make it too busy or full and not able to handle legitimate requests.
U2R	rootkit perl loadmodule buffer-overflow	The attacker in U2R attacks gets access to normal user account on the network system and exploits vulnerability to gain root access to the system.
R2L	ftp-write spy phf guess-passwd imap warezcliecnt wrezmaster multihop	The attacker in R2U attacks send packets to a target machine through a network, then exploits vulnerability to gain local access as a normal user.

In addition, the NSL-KDD dataset includes one more test set, named KDDTest^{-21}, which consists of approximately 11,850 data records. This set contains some new attacks that do not appear in the KDDTrain+ dataset which makes the detection of those attacks even harder.

TABLE 2.3: A list of all features in Kyoto 2006+ dataset

Feature name	Description
1. Duration	Length (number of seconds) of the connection
2. Service	The connection's service type, e.g., http, telnet, etc
3. Source bytes	The number of data bytes sent by the source IP address
4. Destination bytes	The number of data bytes sent by the destination IP address
5. Count	The number of connections to the same host as the current connection in the past two seconds
6. Same-srv-rate	% of connections to the same service in Count feature
7. Serror-rate	% of connections that have "SYN" errors in Count feature
8. Srv-serror-rate	% of connections that have "SYN" errors (same-service connection)
9. Dst-host-count	Among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose source IP address is also the same to that of the current connection
10. Dst-host-srv-count	among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose service type is also the same to that of the current connection
11. Dst-host-same-src-port-rate	% of connections whose source port is the same to that of the current connection in Dst host count feature
12. Dst-host-serror-rate	% of connections that have "SYN" errors in Dst host count feature
13. Dst-host-srv-serror-rate	% of connections that "SYN" errors in Dst-host-srv-count feature
14. Flag	the state of the connection at the time the summary was written (which is usually when the connection terminated). The different states are summarised in the below section.
15. IDS-detection	reflects whether IDS(Intrusion Detection System) triggered an alert for the connection.
16. Malware-detection	indicates whether malware, also known as malicious software, was observed in the connection.
17. Ashula-detection	means whether shellcodes and exploit codes were used in the connection by using the dedicated software.
18. Label	indicates whether the session was attack or not; '1' means the session was normal, '-1' means known attack was observed in the session, and '-2' means unknown attack was observed in the session.
19. Source-IP-Address	indicates the source IP address used in the session.
20. Source-Port-Number	indicates the source port number used in the session.
21. Destination-IP-Address	indicates the source IP address used in the session.
22. Destination-Port-Number	indicates the destination port number used in the session.
23. Start-Time	indicates when the session was started.
24. Duration	indicates how long the session was being established.

2.5.3 Kyoto 2006+ Dataset

Kyoto 2006+ dataset was presented by Song et al. [72]. The dataset covers over three years of real traffic data collected from both honeypots and regular servers that are deployed at Kyoto University. It consists of approximately 50,033,015 normal sessions, 43,043,255 attack sessions and 425,719 sessions which were unknown attacks. Each connection in the dataset is unique with 24 features. Among those features, the authors extracted 14 statistical features from KDD Cup 99 dataset as well as 10 features from their networks. They claim that these features are the most suitable for network IDS. For experimental purposes in this study, all categorical features are converted to binary and normalised.

The dataset generally represents two types of records: normal and attack. In addition, samples belonging to known and unknown attacks are considered as a single attack class, because no much information about the attack types is given in the dataset, and labelled as -1. Table 2.3 shows a list of all features in the Kyoto 2006+ dataset and the detailed description of each feature as it is shown in [72].

2.6 Summary

This chapter has presented an introduction to anomaly detection and briefly discussed some of the existing network anomaly-based detection techniques. Then, it has presented an overview of some benefits of using dependency measures in reducing the false alarm rates of anomaly detection systems. After that, it has discussed the possibility, and advantages of including a feature selection step as part of the detection process. Two main directions of research in feature selection are reviewed:

supervised and unsupervised feature selections. This chapter has investigated some of the limitations of the related feature selection methods and pointed out some possible solutions for overcoming these limitations that will be discussed in some detail in the following chapters. This chapter can be summarised as follows.

Several attempts have been made to develop an efficient dependency measure. One of the most popular measures is the correlation coefficient that has been applied in various domains due to its simplicity and ease of implementation. However, there are two main limitations for linear correlation measures that make them vulnerable to an increasing number of false alarms. Firstly, in the case of the correlation coefficient between two random variables being equal to zero, these two variables are not necessarily independent of each other as has been discussed in Section 2.2 when using this measure. The second weakness is that these methods are not able to extract nonlinear relationships between variables. Mutual information measure addresses some of the linear correlation coefficient deficiencies. MI has been proven to be sensitive to both linear and nonlinear correlations. It provides a solid framework for quantifying the amount of information shared between two random variables. Therefore, applying mutual information to extract the linear and nonlinear correlations between network traffic records and to build anomaly detection systems reduces the high rate of false alarms of these systems.

There is a substantial body of research on supervised feature selection methods based on mutual information. As discussed above, these methods have some shortcomings. For example, there is not a specific formula to select an appropriate value for the parameter β' in Battiti's MIFS, Kwak's MIFS-U and Amiri's MMIFS. In addition, the unbalance between the left hand side and right hand side of the selection criterion provided by NMIFS, mRMR, MMIFS and Vinh's method remains

not fully solved. As a consequence, this may be lead to the selection of irrelevant features in the optimal subset of features.

The research literature on graph-based unsupervised feature selection methods has pointed out two limitations. Firstly, Laplacian score and LGFS methods ignore the redundancy between the selected features. This may lead to select redundant features and affect the classification accuracy. Secondly, E-LGFS, which is an enhanced version of LGF, considers the redundant features among the selected features but has very high computational cost when dealing with high dimensional data and large-scale data.

The following chapters discuss the contributions to the research in detail. The next chapter presents an anomaly detection system based on mutual information and describes how MI can be used to build an efficient detection system with minimum false alarms.

1

Chapter 3

Anomaly Detection System Based on Nonlinear Correlation Measure

Cyber crimes and malicious network activities have posed serious threats to the entire internet and its users. This issue is becoming more critical, as network-based services are more widespread and closely related to the daily life. Current research on network security mainly focuses on developing preventative measures, such as security policies and secure communication protocols. Meanwhile, attempts have been made to protect computer systems and networks against malicious behaviours by using intrusion detection systems. Clearly, the collaboration of IDSs and preventative measures can provide a safe and secure communication environment.

As shown in Chapter 2, a significant amount of work has been conducted to develop intelligent intrusion detection systems. However, one technical challenge, namely reducing false alarms, has been associated with the development of anomaly-based IDSs since 1990s. This may be due to the fact that most of the existing approaches

either ignore the correlations between traffic records or do not take nonlinear correlation into account. Recent literatures on intrusion detection techniques have shown that correlation analysis is one of the effective ways to improve the detection ability and reduce the false alarm rates. Detection systems proposed in [6, 31, 32, 34] are examples of anomaly detection techniques that use different linear correlation measures to develop their systems. However, the false positive rate of these systems is still high. This is because in most communication (for example the real-work communication), the correlation can be linear and nonlinear.

The goal of this chapter is to use a Nonlinear Correlation Coefficient (NCC) based on similarity measure to extract both linear and nonlinear correlations between network traffic records. This extracted information is used in building an anomaly detector that enhances the detection rate with a relatively low false rate. The main objective is the efficient use of mutual information, which provides a theoretical framework for measuring the relationship between two random variables. It is sensitive to both linear and nonlinear correlations and helps to improve the detection accuracy and decrease the rate of false alarms of the proposed detection system. This approach is designed based on the works published in [13] and [14].

The outline of this chapter is as follows. Section 3.1 provides a description of the Linear and nonlinear-based correlation measures. Section 3.2 describes the proposed intrusion detection framework showing the different stages required to build the detection method. Section 3.3 presents the experimental details and results. Finally, a summary to the chapter is drawn in Section 3.4.

3.1 Linear and Nonlinear Correlation Analysis

Measuring the correlation between two random variables has been an active research study over the past decades. It has been applied to solve many statistical problems. Thus, several measures have been proposed in literature. Chapter 2 has introduced the definitions of two of the most popular correlation measures, linear correlation coefficient and mutual information. This section elaborates on the way these measures are used on the proposed detection approach.

3.1.1 Pearson's Correlation Coefficient

Pearson's Correlation Coefficient (PCC) is the most common measure of correlation coefficient, which is sensitive only to a linear relation between two random variables. Given two random variables X and Y, where

$$X = \{x_1, x_2, \ldots, x_N\} \tag{3.1}$$

$$Y = \{y_1, y_2, \ldots, y_N\} \tag{3.2}$$

in which X is a collection of N samples of random variable and Y is a collection of N samples of a second random variable, the PCC of the variables X and Y is defined as follows.

$$PCC(X,Y) = \frac{\sum_{i=1}^{N}(x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{X})^2 \sum_{i=1}^{N}(y_i - \overline{Y})^2}}. \tag{3.3}$$

Equation (3.3) calculates the similarity between variable X and Y, and ranges from 0 to 1, in which a strong relationship means the value of PCC is close to 1 and a weak relationship means a value close to 0. However, as mentioned in Chapter 2, although the linear correlation coefficient is widely applied, it is not totally satisfactory to measure the correlation between random variables as it provides little information about their relationship structure [73]. The mutual information measure has successfully addressed some deficiencies of the linear correlation coefficient. It is able to measure dependence in the presence of a linear and nonlinear structure between the random variables. However, the disadvantage of MI is that it does not range in a definite closed interval [0, 1] as the correlation coefficient does [41]. Therefore, Wang et al. [41] developed a revised version of the MI, named Nonlinear Correlation Coefficient, NCC in short.

3.1.2 Nonlinear Correlation Coefficient

Nonlinear Correlation Coefficient is based on mutual information, which is a quantity measuring the relationship between two random variables. Given the same random variables X and Y, the Nonlinear Correlation Coefficient (NCC), as discussed in Chapter 2, is denoted by Equation (3.4).

$$NCC(X;Y) = 2 + \sum_{i=1}^{b}\sum_{j=1}^{b}\frac{n_{ij}}{N}log_b\frac{n_{ij}}{N}. \tag{3.4}$$

where, n_{ij} is the number of samples distributed in the ijth rank grid, and N is the total number of sample pairs.

For a multi-record scenario, the correlation matrix S of n observed records is used and can be written as

$$S = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix}. \tag{3.5}$$

The elements of the matrix S are the correlation coefficients between distinct pairs of records. The values of elements can be obtained using Equation (3.3) for linear correlation PCC and Equation (3.4) for nonlinear correlation respectively NCC. It is noticed that S is a symmetric matrix and the element values along its diagonal are equal to one; this is because $s_{ij} = s_{ji}$, when $i \neq j$, $1 \leq i \leq n$ and $1 \leq j \leq n$.

3.2 Intrusion Detection Based on Correlation Coefficient

The framework of the proposed intrusion detection system is depicted in Figure 3.1. The detection framework is comprised of four main stages: (1) data collection, where a sequence of network packets is collected; (2) data preprocessing, where training and test data are preprocessed; (3) detection model, where the detection model is trained; and (4) attack recognition, where the trained detection model is used to detect intrusions on the test data.

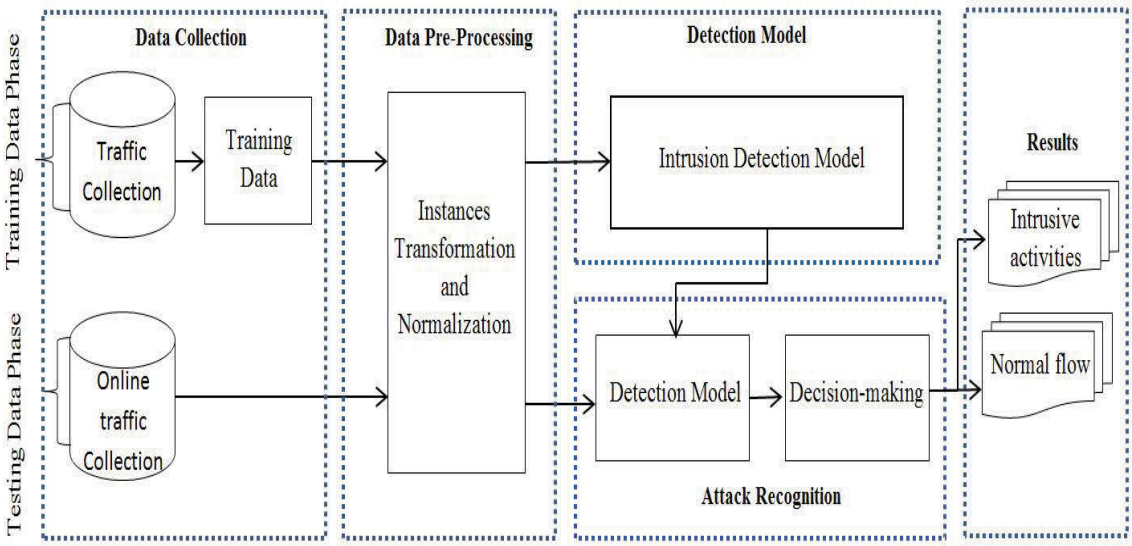

FIGURE 3.1: Overall procedures of the proposed intrusion detection framework

Data Collection:

Data collection is the first and a critical step to intrusion detection. The type of data source and the location where data is collected from are two determinate factors in the design and the effectiveness of an IDS. To provide the best suited protection for the targeted host or networks, this study proposes a network-based IDS in this work. The proposed IDS runs on the nearest router to the victim(s) and monitors the inbound network traffic. During the training stage, the collected data samples are categorised with respect to the transport/internet layer protocols and are labelled against the domain knowledge. However, the data collected in the test stage are categorised according to the protocol types only.

Data Preprocessing:

The data obtained during the phase of data collection are first processed to generate the basic features such as the ones in the NSL-KDD dataset [71]. This phase contains two main steps shown as follows.

(a) **Data transferring:** The detection model requires each record in the input data to be represented as a vector of real number value. Thus, every symbolic feature in a dataset is first converted into a numerical value. For example, the NSL-KDD dataset contains numerical as well as symbolic features. These symbolic features include the type of protocol that is TCP, UDP and ICMP, service type for example, HTTP, FTP, Telnet and TCP status flag for example, SF, REJ. The method simply replaces the values of the categorical attributes with numeric values.

(b) **Data normalisation:** An essential step of data preprocessing after transferring all symbolic attributes into numerical values is normalisation. Data normalisation is a process of scaling the value of each attribute into a well-proportioned range, so that the bias in favor of features with greater values is eliminated from the dataset. Data used in Section 3.3 are standardized. Every feature within each record is normalised by the respective maximum value and falls into the same range of [0-1]. The transferring and normalisation process will also be applied for test data.

Detection Model:

Once the training dataset is preprocessed, this data is then taken into the detection model phase to build the intrusion detection model. This model is then used to distinguish Normal data from non-Normal. It is to be noticed that, in order to train the detection model, two main components are required: normal profile and a pre-defined threshold σ.

To determine the similarity between the normal record and a new incoming record, the detection method is divided into two different stages. Firstly, normal profile is built for normal records and the mean value of correlation coefficient among the normal ones is obtained. Secondly, a threshold value is used to determine whether the new incoming record is normal or not. The following subsections describe the technique that is used to create the normal profile and select the threshold value.

(a) **Normal Profile Generation Using NCC:** Given a set of n normal training traffic samples $X^{normal} = \{x_1^{normal}, x_1^{normal}, \ldots, x_n^{normal}\}$, the NCC is first calculated using Equation (3.4), between the n normal records and then the correlation matrix S using Equation (3.5) for the normal records is generated. After that, the mean $\overline{S}_c$ of each column c in S is defined as

$$\overline{S}_c = \frac{1}{n}\sum_{i=1}^{n} S_{ic} = \frac{1}{n}\sum_{i=1}^{n} NCC_{ic}, \tag{3.6}$$

where NCC_{ic} represents the NCC between record i and record c and n is the number of rows in each column of matrix S.

Equation (3.6) shows the mean value of the NCC values in each column in the matrix S. Finally, the mean value of the results obtained using Equation (3.6) is calculated using Equation (3.7) and denoted by $\overline{NCC^n}$.

$$\overline{NCC^n} = \frac{1}{n}\sum_{c=1}^{n}\overline{S}_c \tag{3.7}$$

(b) **Threshold Selection:** The selection of the threshold value σ is a delicate task when designing IDS. It directly influences the False Positive Rate (FPR) and Detection Rate (DR). In other words, a larger value of threshold generates less FPR and a smaller value of threshold leads to higher DR.

In fact, the key point of both PCC measure and NCC measure, as explained in Section 3.1, is measuring the correlation between two random variables. To the best of our knowledge, there is no exact mathematical solution to determine the threshold value as the degree of strong or weak correlations. Therefore, during the training phase various values for the threshold range from 0 to 1 have been tested. The experimental result shows that a larger threshold value leads to less false positive alarms but less DR as well. Therefore, empirically the threshold values between 0.1 and 0.5 give high DRs and low FPR. More explanation about the threshold selection is given in Section 3.3.

Attack Recognition:

After completing the whole iteration process, the final detection model can be determined which can differentiate between the normal and intrusion traffics using

the saved trained model. The test data is then directed to the saved trained detection model to detect intrusions. If the detection model confirms that the record is abnormal, an alarm will be sent to the administrator indicating that there is an attack.

Detection Algorithm: Similar to the normal profile development process, for any new incoming record $n+1$, the $NCC^{n,n+1}$ between the new incoming record and the records in the normal profile is calculated using Equation (3.4). Then, an $n \times 1$ matrix $G_{n+1} = [G_{i(n+1)}]_{n\times 1}$, where $G_{i(n+1)}$ stands for the correlation between the $n+1$ (i.e., the new) record and the i-th record in the normal profile, is generated. After that, the mean of G_{n+1}, denoted by $\overline{G}_{n+1}$ is calculated by

$$\overline{G}_{n+1} = \frac{1}{n}\sum_{i=1}^{n} G_{i(n+1)} = \frac{1}{n}\sum_{i=1}^{n} NCC_i^{n,n+1} \tag{3.8}$$

where $NCC_i^{n,n+1} = G_{i(n+1)}$ represents the NCC between record $n+1$ and the i-th record in the normal profile.

Let us denote the $\overline{G}_{n+1}$ obtained above by $\overline{NCC^{n,n+1}}$. After that, the difference between the mean of the normal profile given in Equation (3.7) and the mean in Equation (3.8) is calculated by

$$\mid \overline{NCC^{n}} - \overline{NCC^{n,n+1}} \mid . \tag{3.9}$$

Finally, the incoming record is considered as an attack or abnormal if the difference between $\overline{NCC^{n}}$ and $\overline{NCC^{n,n+1}}$ is greater than a pre-defined threshold σ or not.

The flow chart given in Figure 3.2 illustrates the aforementioned processes of the detection algorithm. In the case of developing an intrusion detection model based on the PCC measure, similar processes to the above detection algorithm are applied. The comparison results of both NCC intrusion detection system and PCC intrusion detection system are given in the next section. Both systems are evaluated in terms of false positive rate and detection rate.

3.3 Experimental Results and Analysis

This section describes the results obtained by applying the proposed intrusion detection model in Section 3.2 to detect the normal records and six different types of DoS attacks.

3.3.1 Dataset Selection

In this experimentation, NSL-KDD dataset (http://iscx.ca/NSL-KDD), which is an enhanced version of KDD Cup 99 dataset, is utilised to demonstrate the effectiveness of the proposed approach. As shown in Chapter 2, the KDD Cup 99 dataset has some problems such as including a very large number of redundant records, which affect the performance of intrusion detection systems. Thus, Tavallaee et al. in [71] addressed these limitations and presented a new revised version of KDD Cup 99 termed the NSL-KDD. More details about both KDD Cup 99 and NSL-KDD can be found in Chapter 2, Section 2.5.

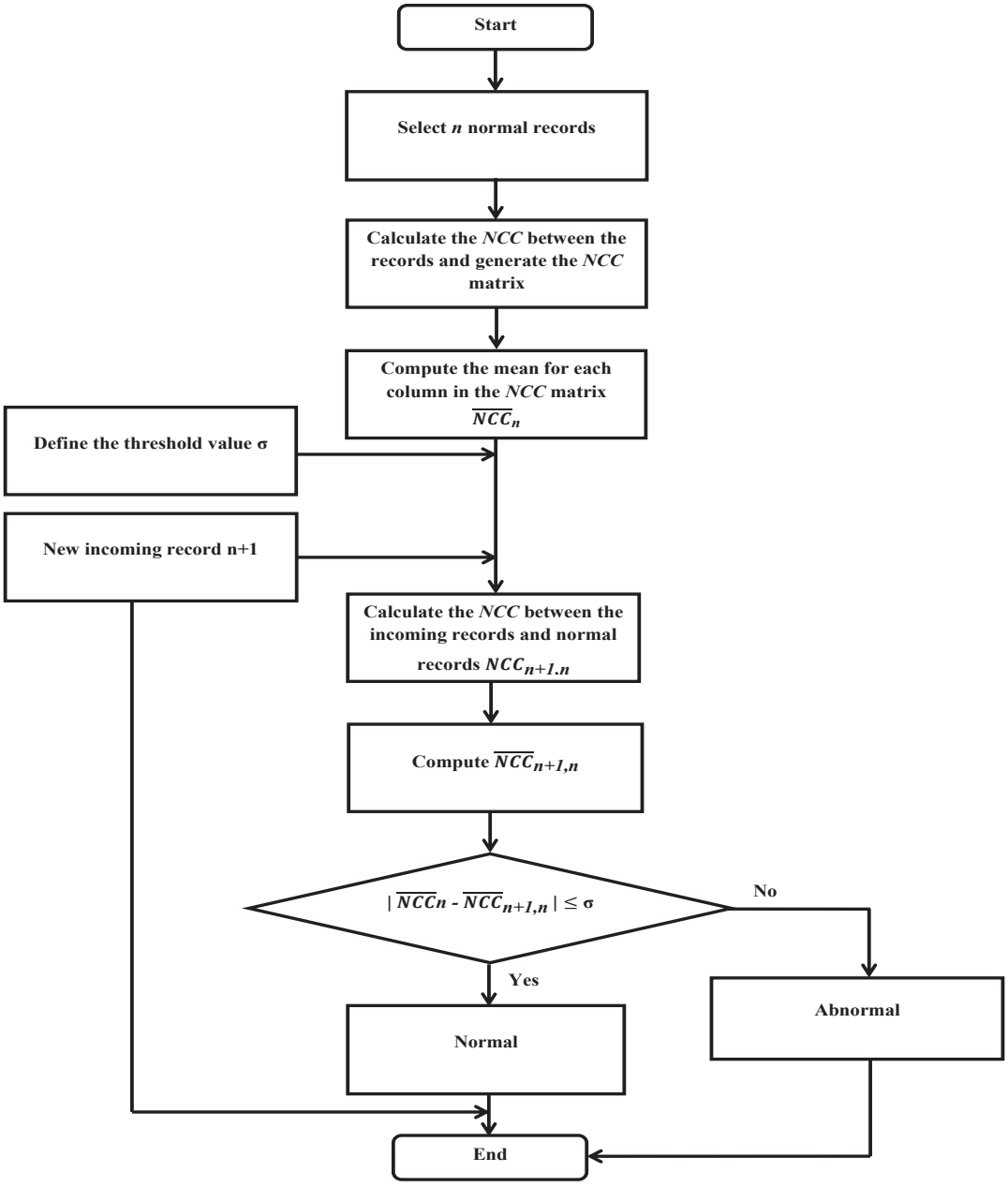

FIGURE 3.2: The flow chart of the proposed algorithm

For training and testing purposes, six types of DoS attacks including smurf, neptune, land, teardrop, back and pod attacks are randomly selected to train and test the detection system. The distribution of records of various types in training and testing phases are listed in Table 3.1 and Table 3.2 respectively.

TABLE 3.1: Sample distribution on the training dataset

Normal	Attack						Total
	Neptune	*Land*	*Smurf*	*Teardrop*	*Back*	*Pod*	
	590	19	251	169	365	162	
1980			1556				3536

TABLE 3.2: Sample distribution on the testing dataset

Normal	Attack						Total
	Neptune	*Land*	*Smurf*	*Teardrop*	*Back*	*Pod*	
	1840	19	1566	1313	988	1761	
14590			7487				22077

3.3.2 Performance Evaluation

The performance of intrusion detection technique is defined by its ability to make correct predictions. Comparing an event with the predictions from the IDS, there are four possible outcomes, as shown in Table 3.3. These outcomes are known as the confusion matrix. Several experiments have been conducted to examine the performance and effectiveness of the proposed detection system in terms of the Detection Rate (DR) and False Positive Rate (FPR). The DR represents the

capability of IDS in detecting attacks, while the FPR refers to the probability of IDS triggering an alarm when there is no attack occurring. The definition of the detection rate and false positive rate are given by Equation (3.10) and Equation (3.11) respectively.

$$DetectionRate = \frac{TP}{TP + FN} = \frac{\#correct\ intrusions}{\#intrusions}, \tag{3.10}$$

$$FalsePositiveRate = \frac{FP}{FP + TN} = \frac{\#intrusions\ as\ normal}{\#intrusions}, \tag{3.11}$$

where

- True Positive (TP) is the number of actual attacks classified as attacks,
- True Negative (TN) is the number of actual normal records classified as normal ones,
- False Positive (FP) is the number of actual normal records classified as attacks, and
- False Negative (FN) is the number of actual attacks classified as normal records.

3.3.3 Results and Discussion

During the training phase, both NCC measure and PCC measure are applied. By following the proposed detection algorithm shown in Figure 3.2, the correlation coefficients (s_{ij}) between the selected records is calculated to generate the normal profile

Table 3.3: Confusion matrix

Actual	Prediction	
	Normal	*Attack*
Normal	TP	FN
Attack	FP	TN

using both PCC measure and NCC measure respectively. Figure 3.3 illustrates the two different correlation matrices S^{PCC} and S^{NCC} of normal profiles for the same samples. Each element s_{ij} in the matrices describes the correlation between the i^{th} and j^{th} records.

However, to differentiate normal and abnormal records, it is necessary to define the pre-defined sensitive threshold σ firstly. To the best of our knowledge, there is no good way to solve this value theoretically. Hence, the conventional method is adopted to determine this value by setting different values for the threshold. The value varies from 0.1 to 0.5 with the step length 0.1, and the results obtained by each value are discussed in Table 3.4 and Figure 3.4. The results in Table 3.4 and Figure 3.4 are based on the detection rate and the false positive rate respectively. It can be seen from the obtained results that the detection performance of the system completely depends on the value of the threshold σ. For example, good detection results are achieved when the value of σ is neither too small nor too large, such as $\sigma = 0.2$ or 0.3.

From the comparison between the various threshold values and results illustrated in Table 3.4 and Figure 3.4, when $\sigma = 0.3$, the DR for normal records decreases slightly from 100% to when $\sigma = 0.5$ with a DR of 99.85%. Another example is for Neptune attacks, the DR is equal to 98.81% when $\sigma = 0.5$ and is equal to 99.66%

$$S^{PCC} = \begin{pmatrix}
1.0000 & 0.8972 & 0.8892 & 0.9548 & 0.9841 & 0.4312 & 0.3832 & 0.4599 & 0.8789 & 0.7032 & 0.6321 & 0.6721 \\
0.8972 & 1.0000 & 0.9997 & 0.7694 & 0.8131 & 0.0320 & 0.0536 & 0.0434 & 0.6265 & 0.4233 & 0.3632 & 0.3771 \\
0.8892 & 0.9997 & 1.0000 & 0.7623 & 0.8023 & 0.0215 & 0.0337 & 0.0278 & 0.6147 & 0.4130 & 0.3445 & 0.3560 \\
0.9548 & 0.7694 & 0.7623 & 1.0000 & 0.9636 & 0.6152 & 0.4005 & 0.6289 & 0.9340 & 0.8142 & 0.6016 & 0.6388 \\
0.9841 & 0.8131 & 0.8023 & 0.9636 & 1.0000 & 0.5677 & 0.4622 & 0.5998 & 0.9388 & 0.7915 & 0.6777 & 0.7401 \\
0.4312 & 0.0320 & 0.0215 & 0.6152 & 0.5677 & 1.0000 & 0.4942 & 0.9517 & 0.7553 & 0.8314 & 0.4528 & 0.5442 \\
0.3832 & 0.0536 & 0.0337 & 0.4005 & 0.4622 & 0.4942 & 1.0000 & 0.5443 & 0.3967 & 0.2477 & 0.9494 & 0.9208 \\
0.4599 & 0.0434 & 0.0278 & 0.6289 & 0.5998 & 0.9517 & 0.5443 & 1.0000 & 0.8017 & 0.8756 & 0.5039 & 0.6145 \\
0.8789 & 0.6265 & 0.6147 & 0.9340 & 0.9388 & 0.7553 & 0.3967 & 0.8017 & 1.0000 & 0.9515 & 0.5528 & 0.6586 \\
0.7032 & 0.4233 & 0.4130 & 0.8142 & 0.7915 & 0.8314 & 0.2477 & 0.8756 & 0.9515 & 1.0000 & 0.3461 & 0.4842 \\
0.6321 & 0.3632 & 0.3445 & 0.6016 & 0.6777 & 0.4528 & 0.9494 & 0.5039 & 0.5528 & 0.3461 & 1.0000 & 0.9763 \\
0.6721 & 0.3771 & 0.3560 & 0.6388 & 0.7401 & 0.5442 & 0.9208 & 0.6145 & 0.6586 & 0.4842 & 0.9763 & 1.0000
\end{pmatrix}$$

(A) PCC-based correlation matrix

$$S^{NCC} = \begin{pmatrix}
1.0000 & 0.9696 & 0.8087 & 0.8087 & 0.8441 & 0.7341 & 0.6166 & 0.6493 & 0.5764 & 0.5426 & 0.7341 & 0.7341 \\
0.9696 & 1.0000 & 0.9696 & 0.9696 & 0.9339 & 0.8424 & 0.7744 & 0.7249 & 0.6633 & 0.6166 & 0.8424 & 0.6633 \\
0.8087 & 0.9696 & 1.0000 & 0.8087 & 0.8441 & 0.7341 & 0.8326 & 0.6493 & 0.5764 & 0.5426 & 0.7341 & 0.5764 \\
0.8087 & 0.9696 & 0.8087 & 1.0000 & 0.8441 & 0.7341 & 0.6166 & 0.6493 & 0.5764 & 0.5426 & 0.7341 & 0.5764 \\
0.8441 & 0.9339 & 0.8441 & 0.8441 & 1.0000 & 0.7249 & 0.7041 & 0.7530 & 0.7249 & 0.8148 & 0.9339 & 0.9339 \\
0.7341 & 0.8424 & 0.7341 & 0.7341 & 0.7249 & 1.0000 & 0.7744 & 0.9339 & 0.6633 & 0.6166 & 0.6633 & 0.6633 \\
0.6166 & 0.7744 & 0.8326 & 0.6166 & 0.7041 & 0.7744 & 1.0000 & 0.7041 & 0.7744 & 0.5463 & 0.7744 & 0.5747 \\
0.6493 & 0.7249 & 0.6493 & 0.6493 & 0.7530 & 0.9339 & 0.7041 & 1.0000 & 0.9339 & 0.8148 & 0.7249 & 0.7249 \\
0.5764 & 0.6633 & 0.5764 & 0.5764 & 0.7249 & 0.6633 & 0.7744 & 0.9339 & 1.0000 & 0.7716 & 0.8424 & 0.6633 \\
0.5426 & 0.6166 & 0.5426 & 0.5426 & 0.8148 & 0.6166 & 0.5463 & 0.8148 & 0.7716 & 1.0000 & 0.6166 & 0.7716 \\
0.7341 & 0.8424 & 0.7341 & 0.7341 & 0.9339 & 0.6633 & 0.7744 & 0.7249 & 0.8424 & 0.6166 & 1.0000 & 0.6633 \\
0.7341 & 0.6633 & 0.5764 & 0.5764 & 0.9339 & 0.6633 & 0.5747 & 0.7249 & 0.6633 & 0.7716 & 0.6633 & 1.0000
\end{pmatrix}$$

(B) NCC-based correlation matrix

FIGURE 3.3: Matrices expressions of two different measures for normal profiles, (a) PCC-based correlation matrix (b) NCC-based correlation matrix

when $\sigma = 0.3$. In addition, even though there is a slight difference in the DR when $\sigma = 0.3$ and $\sigma = 0.1$ in some cases, as shown in Table 3.4, the FPRs, shown in Figure 3.4, when $\sigma = 0.1$ is obviously higher than when $\sigma = 0.3$. To sum up, it was to optimise performance that we choose $\sigma = 0.3$ as a fixed threshold value for the

proposed detection model.

During the test process, the mean correlation coefficient $\overline{NCC}_{n+1,i}$ among each new record and the corresponding normal profile which is built based on the normal traffic records is calculated. If the distance between the mean coefficient of normal profile and $\overline{NCC}_{n+1,i}$ exceeds the pre-defined threshold 0.3, it would be treated as an abnormal record. Otherwise it would be considered as legitimate traffic.

TABLE 3.4: DRs for various threshold values on the training dataset

Type of	Attack				
records	0.1	0.2	0.3	0.4	0.5
Normal	100%	100%	100%	99.94%	99.85%
Teardrop	100%	99.41%	99.41%	98.22%	94.08%
Smurf	100%	99.60%	99.60%	98.41%	97.61%
Pod	100%	100%	100%	100%	98.76%
Neptune	99.83%	99.66%	99.66%	99.32%	98.81%
Back	99.45%	99.45%	99.18%	98.35%	97.26%
Land	100%	100%	100%	95%	95%

Considering the selected threshold value $\sigma = 0.3$, the confusion matrix presented in Table 3.5 shows that the intrusion detection algorithm using NCC measure achieves high accuracy in detecting both normal records (99.84%) and attack records (99.55%).

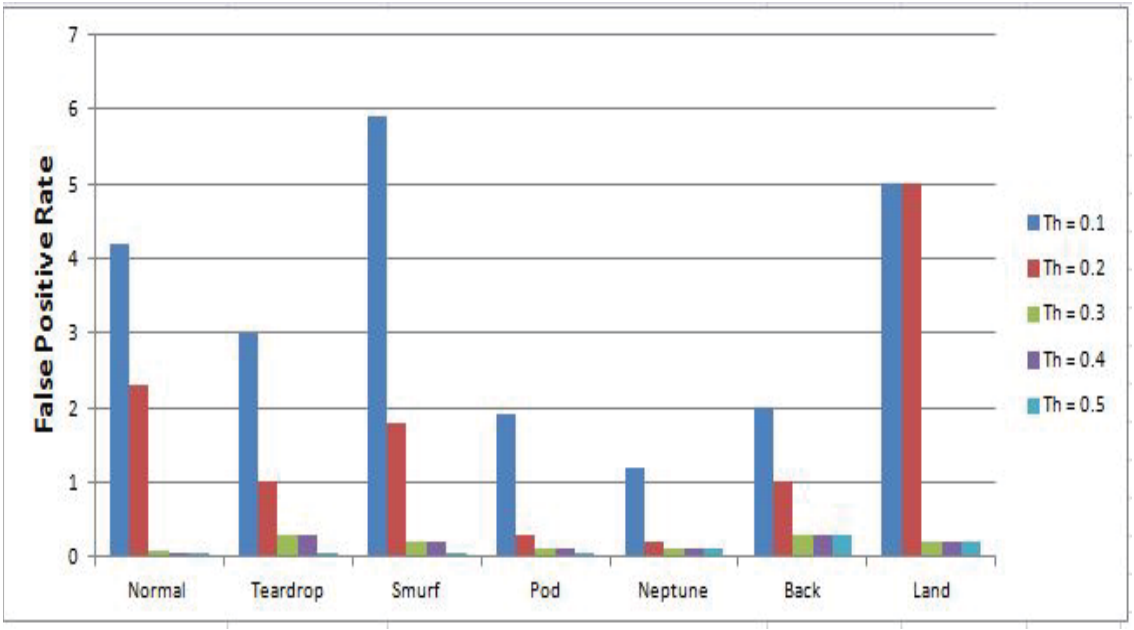

FIGURE 3.4: FPRs for various threshold values on the training dataset

TABLE 3.5: Confusion matrix for NCC-training set

Predicted actual	Normal	Attack	Correct
Normal	1977	3	99.84%
Attack	7	1549	99.55%

3.3.4 Comparative Study

Considering the selected threshold value $\sigma = 0.3$, the results presented in Table 3.6 show that during the testing phase the proposed detection system achieves a detection rate of 98.754% and a false positive rate of 1.246%, which is better than the DR (of 97.632%) and FPR (of 2.367%) achieved by the PCC measure. This can be attributed to the fact that the proposed model utilises both linear and nonlinear correlations between network traffic records when detecting attacks.

In addition, compared to some of the state-of-the-art systems, the DR of the NCC-based IDS outperforms the DR obtained by other proposed methods. More importantly, the FPR of NCC-based IDS also performs better than other existing methods.

TABLE 3.6: Comparison of detection and false alarm between different IDS using NSL-KDD dataset

Methods	False positive rates (%)	Detection rates (%)
NCC-MI (proposed method)	1.246	98.754
PCC	2.367	97.632
Naïve Bayes Tree [74]	2.0	**
SVM [75]	14	93.4
DM – Naïve Bayes [76]	3.0	96.5

Note: ** Indicates data not provided by the authors in their paper.

Additionally, given the columns in PCC and NCC matrices are S_j^{PCC} and S_j^{NCC} respectively, the covariance of these two columns is used to illustrate the significant difference as shown in Equation (3.12).

$$Cov(S_j^{PCC}, S_j^{NCC}) = E[(S_j^{PCC} - ES_j^{PCC})(S_j^{NCC} - ES_j^{NCC})] \tag{3.12}$$

It should be noticed that the correlation coefficient matrices are symmetric. Therefore, the dimension of columns for which the need to calculate the covariance decreases gradually.

3.4 Summary

This chapter has utilized a Nonlinear Correlation Coefficient (NCC) measure to quantitatively measure the linear and nonlinear relations between two random variables. NCC is designed based on mutual information. An intrusion detection system based on the assumption that intrusion behaves differently from normal network traffic is proposed. To equip the intrusion detection model with high detection performance in recognising the deviation of an attack from the normal traffic flow, the NCC is adopted into the proposed detection model to extract the correlation between network traffic records. This makes the algorithm feasible in not only linear correlation extraction but also nonlinear correlation extraction.

The findings is verified by experimentation and comparison with PCC measure. The experimental results have shown that a NCC-based intrusion detection system achieves not only lower FPR but also higher DR than those of a PCC-based intrusion detection algorithm. Furthermore, the performance of the proposed model outperforms some of the existing IDSs.

However, the proposed intrusion detection scheme still needs to be further studied in some aspects. For example, in general, we need to develop an IDS to deal with a large volume of network traffic data which has a very high computational cost. Chapter 4 and Chapter 5 will address this problem and propose solutions to reduce the computational complexity of the proposed detection model. In addition, more sophisticated classification techniques will be employed, such as machine learning methods, in the future work to improve the classification accuracy of the IDS and alleviate the false positive rate.

Chapter 4

Supervised Filter-based Feature Selection Algorithm for IDS

Along with fast improvement of data acquisition systems, large-scale data are easy to accumulate. A large amount of data usually causes many mathematical difficulties which then leads to higher computational complexity. It needs large storage space and a long time for training and testing processes. This problem raises a major challenge to intrusion detection systems, which need to examine all features in the data to identify intrusive patterns. To solve this problem, feature selection becomes an important part of most IDS applications. This technique aims to remove noisy and redundant features from the data by selecting a subset of the most important features to the classification purposes.

As shown in section 2.3.2 of Chapter 2, several filter-based feature selection algorithms have been proposed in literature based on the principle of mutual information. Battiti's MIFS [57] was one of the earliest methods that evaluates features

based on their relevance to the classification, by maximising the information that feature f_i carries about the output class, corrected by subtracting a quantity proportional to the average mutual information with the features that have been selected previously. One can find more details about Battiti's MIFS [57] in Chapter 2. Numerous studies have been conducted to improve Battiti's MIFS including those in [42, 53, 58, 59]. The enhancements in all of these methods have been made on the right-hand side of Battiti's MIFS [57]. However, these methods present some limitations. For example, there is not a specific guideline to select an appropriate value for the parameter β in MIFS [57], MIFS-U [58] and MMIFS [1], where β is a user-defined parameter that is applied to account for the redundancy between the candidate feature and the set of selected features. In addition, the unbalance between the left and right hand sides of the selection criterion in all proposed methods has not been completely solved.

This chapter presents a new filter-based feature selection method, in which theoretical analysis of mutual information is introduced to evaluate the dependence between features and output classes. The most relevant features are retained and used to construct classifiers for respective classes. This will help the classifier to shorten the training and testing time as well as to enhance the classification accuracy. Due to the generality of the proposed algorithm, its flexibility allows it to be applied in various domains, thus we name it Flexible Mutual Information based Feature Selection (FMIFS). As an enhancement of Mutual Information Feature Selection (MIFS) [57] and Modified Mutual Information-based Feature Selection (MMIFS) [1], the proposed feature selection method does not have any free parameter, such as β in MIFS and MMIFS. Therefore, its performance is free from being influenced by any inappropriate assignment of value to a free parameter and can be guaranteed. Its

effectiveness is evaluated in the cases of network intrusion detection. An Intrusion Detection System (IDS), named Least Square Support Vector Machine based IDS (LSSVM-IDS), is built using the features selected by our proposed feature selection algorithm. This approach is designed based on the work in [15] submitted to IEEE Transaction on Computers.

The outline of this chapter is as follows. Section 4.1 introduces the proposed feature selection algorithm FMIFS. Section 4.2 briefly describes the concept of LS-SVM and details the detection framework showing the additional stages involved in the proposed scheme. Section 4.3 presents the experimental details and results. Finally, a summary to the chapter is drawn in Section 4.4.

4.1 Filter-based Feature Selection

If one considers correlations between network traffic records to be linear associations, then a linear measure of dependence such as linear correlation coefficient can be used to measure the dependence between two random variables. However, considering the real world communication, the correlation between variables can be nonlinear as well. Apparently, a linear measure cannot reveal the relation between two nonlinearly dependent variables. Thus, we need a measure capable of analysing the relation between two variables no matter whether they are linearly or nonlinearly dependent. For these reasons, this work intends to explore a means of selecting optimal features from a feature space regardless of the type of correlation between them.

4.1.1 Flexible Mutual Information based Feature Selection

To remove the burden of setting an appropriate value for β as it is required in Battiti's MIFS [57] and Amiri's MMIFS [1], a new variation of MIFS is proposed in this section. This new feature selection approach suggests an enhancement to the feature selection criterion involved in the computation of the right-hand side of MIFS algorithm as shown in Equation (2.26). Equation (4.1) below recalls Equation (2.26) used for setting the criterion for feature selection of MIFS.

$$J_{MIFS} = I(C; f_i) - \beta \sum_{f_s \in S} I(f_i; f_s). \tag{4.1}$$

where β is the redundancy parameter. The term $I(C; f_i)$ is the amount of information that feature f_i carries about the class C. The term $\beta \sum_{f_s \in S} I(f_i; f_s)$ estimates the redundancy of the ith feature with respect to the subset of previously selected features.

Equation (4.2) shows a new formulation to the feature selection criterion involved, which is intended to determine a feature that maximises the term in Equation (4.2).

Given a training dataset $T = D(F, C)$ with n features and m instances, where $F = \{f_1, ..., f_n\}$ and $D = \{i_1, ..., i_m\}$ are the sets of features and instances, respectively, $C = \{c_1, ..., c_l\}$ represents the set of classes (or labels) which instances belong to. The task is to select the best subset of features $S = \{s_1, s_2, ..., s_{|S|}\}$, where $|S|$ is the number of selected features. The scheme proposed in Equation (4.2) is to select a feature from an initial input feature set that maximises $I(C; f_i)$, which measures the relevance of the feature to the output class, and minimises the average of redundancy MRs simultaneously.

$$G_{MI} = \underset{f_i \in F}{\operatorname{argmax}}(I(C; f_i) - \frac{1}{\mid S \mid} \sum_{f_s \in S} MR), \tag{4.2}$$

MR, in Equation (4.2), is the relative minimum redundancy of feature f_i against feature f_s and is given by Equation (4.3).

$$MR = \frac{I(f_i; f_s)}{I(C; f_i)} \tag{4.3}$$

where $f_i \in F$ and $f_s \in S$. In the case of $I(C; f_i) = 0$, feature f_i can be discarded without computing Equation (4.2). If f_i and f_s are relatively highly dependent with regard to $I(C; f_i)$, feature f_i will contribute redundancy. This is because f_i and C are proven to be independent. Thus, to reduce the number of features that needs to be examined in order to select the optimal number of features, a numerical threshold $Th(= 0)$ value is applied to G_{MI} in (4.2) so that G_{MI} has the following properties:

1. If $(G_{MI} = 0)$, then the current feature f_i is irrelevant or unimportant to the output C because it cannot provide any additional information to the classification after selecting the S subset of features. Thus, the current candidate f_i should be removed from S.

2. If $(G_{MI} > 0)$, then the current feature f_i is relevant or important to the output C because it can provide some additional information to the classification after selecting the S subset of the feature. Thus, the current candidate f_i should be added into S.

3. If ($G_{MI} < 0$), then the current feature f_i is redundant to the output C because it can cause reduction in the amount of MI between the selected subset S and the output C. It is worth noting that the right hand term in Equation (4.2), which measures the redundancy among features, is larger than the left hand term, which measures the relevancy between feature f_i and the output class. Thus, feature f_i should be removed from S.

The selection process of FMIFS is given by Algorithm 1.

Algorithm 1 Flexible mutual information based feature selection

Input: Feature set $F = \{f_i, i = 1, ..., n\}$

Output: S - the selected feature subset

begin

Step1. Initialization: set $S = \phi$

Step2. Calculate $I(C; f_i)$ for each feature, $i = 1, ..., n$

Step3. $n_f = n$; Select the feature f_i such that:

$$\underset{f_i}{\operatorname{argmax}}(I(C; f_i)), i = 1, ..., n_f,$$

Then, set $F \leftarrow F \backslash \{ f_i \}$; $S \leftarrow S \cup \{ f_i \}$; $n_f = n_f - 1$.

Step4. while $F \neq \phi$ **do**

Calculate G_{MI} in (4.2) to find f_i where $i \in \{1, 2, \cdots, n_f\}$;

$n_f = n_f - 1$;

$F \leftarrow F \backslash \{f_i\}$;

if *(*$G_{MI} > 0$*)* **then**

$S \leftarrow S \cup \{ f_i \}$.

end

end

Step 5. Sort S according to the value of G_{MI} of each selected feature.

return S

4.1.2 Feature Selection Based on Linear Correlation Coefficient

In order to demonstrate the flexibility and effectiveness of FMIFS against feature selection based on linear dependence measure, we substitute MI by Linear Correlation Coefficient (LCC) in Algorithm 1.

As discussed in Chapter 2, Linear Correlation Coefficient (LCC) [77] is one of the most popular dependence measures evaluating the relationship between two random variables. Whilst LCC is fast and accurate in measuring the correlations between random linearly dependent variables, it is insensitive to nonlinear correlations. Given that two random variables $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$, where n is the total number of samples, the correlation coefficient between these two variables is defined in Equation (4.4).

$$corr(X;Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}. \tag{4.4}$$

The value of *corr* $(X;Y)$ falls in a definite closed interval [-1,1]. A value close to either -1 or 1 indicates a strong relationship between the two variables. Otherwise, the value infers a weak relationship between them. The following shows a different feature selection algorithm based on LCC, and this algorithm is called Flexible Linear Correlation Coefficient Feature Selection (FLCFS). Algorithm 2 is designed to select a feature that maximises G_{corr} in Equation (4.5) and to eliminate irrelevant and redundant features.

$$G_{corr} = \underset{f_i \in F}{\text{argmax}}(corr(C; f_i) - \frac{1}{\mid S \mid} \sum_{f_s \in S} \frac{corr(f_i; f_s)}{corr(C; f_i)}). \tag{4.5}$$

Equation (4.5) is based on the definition of the linear correlation coefficient. The left-hand side of Equation (4.5) calculates the correlation between the candidate feature and the class, while the right-hand side is to eliminate the redundancy between candidate feature and the previously selected features.

Algorithm 2 Flexible Linear Correlation Coefficient based Feature Selection

Input: Feature set $F = \{f_i, i = 1, ..., n\}$

Output: S - the selected feature subset

begin

Step1. Initialization: $S = \phi$

Step2. Calculate $corr(C; f_i)$ for each feature, $i = 1, ..., n$

Step3. $n_f = n$; Select the feature f_i such that:

$$\underset{f_i}{\text{argmax}}(corr(C; f_i)), i = 1, ..., n_f,$$

Then, set $F \leftarrow F \backslash \{ f_i \}$; $S \leftarrow S \cup \{ f_i \}$; $n_f = n_f - 1$.

Step4. while $F \neq \phi$ **do**

Calculate G_{corr} in (4.5) to find f_i where $i \in \{1, 2, \cdots, n_f\}$;

$n_f = n_f - 1$;

$F \leftarrow F \backslash \{f_i\}$;

if *(*$G_{corr} > 0$*)* **then**

$S \leftarrow S \cup \{ f_i \}$.

end

end

Step 5. Sort S according to the value of *Gcorr* of each selected feature.

return S

4.2 Intrusion Detection Framework-based on Least Square Support Vector Machine

The previous chapter introduced the proposed detection framework and described the four different stages involved in the intrusion detection system. These stages are: (1) data collection, sequences of network packets are collected; (2) data preprocessing, where training and test data are preprocessed; (3) classifier training, where the training data is trained for classification problem; and (4) attack recognition, where the classifier is trained using LS-SVM to detect intrusions on the test data.

The proposed framework in this chapter provides an enhancement to the previous proposed framework by including a feature selection step as part of the preprocessing stage and utilising a machine learning method (for example, LSSVM) to classify normal traffic and attacks. The detection framework is depicted in Figure 4.1.

For experimental purposes and to make a fair comparison with those systems that have been evaluated on different types of attacks included in the KDD Cup 99 dataset, five different classes are constructed. One of these classes contains purely the normal records and the other four hold different types of attacks (such as, DoS, Probe, U2R, R2L), respectively. More details about these attacks can be found in Chapter 2.

An essential step of the data preprocessing stage after transferring all symbolic attributes into numerical values and scaling the value of each attribute into a well-proportioned range is feature selection. Even though every connection in a dataset is represented by various features, not all of these features are needed to build an IDS. Therefore, it is important to identify the most informative features of traffic

data to achieve higher performance. In the previous section using Algorithm 1, a flexible method for the problem of feature selection, FMIFS, is developed. However, the proposed feature selection algorithm can only rank features in terms of their relevance but they cannot reveal the best number of features that are needed to train a classifier. Therefore, this study applies the same technique proposed in [1] to determine the optimal number of required features. To do so, the technique first utilises the proposed feature selection algorithm to rank all features based on their importance to the classification processes. Then, incrementally the technique adds features to the classifier one by one. The final decision of the optimal number of features in each method is taken once the highest classification accuracy in the training dataset is achieved. This technique is also applied for MIFS and FLCFS. The selected features for all datasets are depicted in Table 4.1 [A-C], where each row lists the number and the indexes of the selected features with respect to the corresponding feature selection algorithm. In addition, for KDD Cup 99, the proposed feature selection algorithm is applied for the aforementioned classes. The selected features are shown in Table 4.3.

As a classification method to classify normal traffic from abnormal traffic, least squares support vector machine method is applied in this study. Support vector machine is a supervised learning method [78]. It studies a given labelled dataset and constructs an optimal hyperplane in the corresponding data space to separate the data into different classes. Instead of solving the classification problem by quadratic programming, Suykens and Vandewalle [79] suggested re-framing the task of classification into a linear programming problem. They named this new formulation the Least Squares SVM (LS-SVM). LS-SVM is a generalized scheme for classification and also incurs low computation complexity in comparison with the

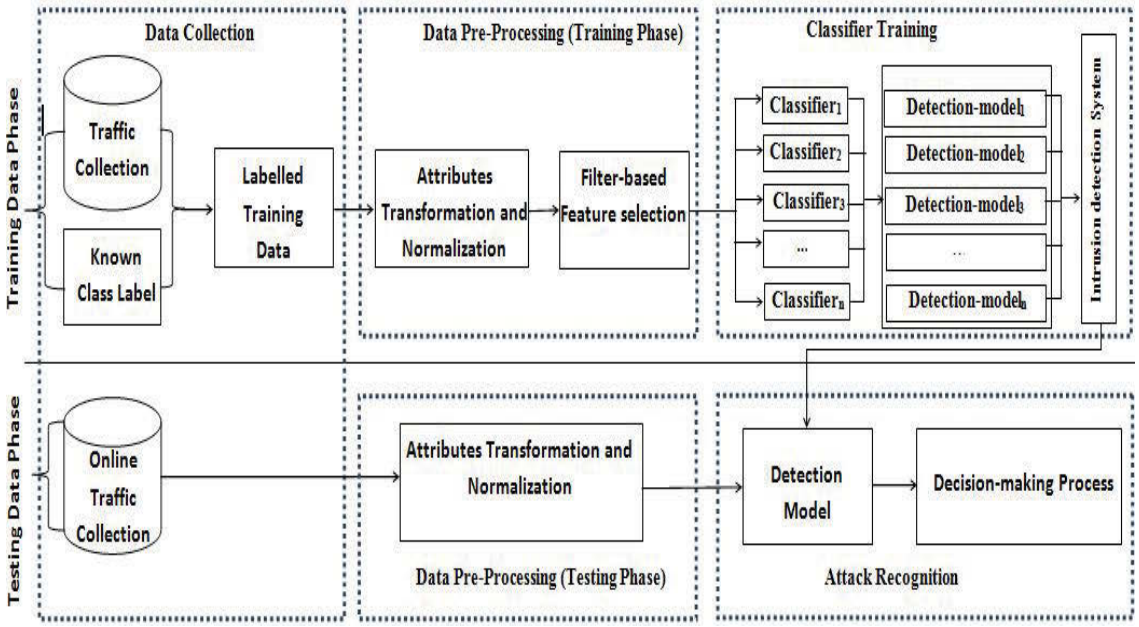

FIGURE 4.1: The framework of the LS-SVM-based intrusion detection system

ordinary SVM scheme [80]. One can find more details about calculating LS-SVM in Appendix A.

Since SVMs can only handle binary classification problems and because for KDD Cup 99 five optimal feature subsets are selected for all classes, five LS-SVM classifiers need to be employed. Each classifier distinguishes one class of records from the others. For example the classifier of Normal class distinguishes Normal data from non-Normal (All types of attacks). The DoS class distinguishes DoS traffic from non-DoS data (including Normal, Probe, R2L and U2R instances) and so on. The five LS-SVM classifiers are then combined to build the intrusion detection model to distinguish all different classes.

In general, it is simpler to build a classifier to distinguish between two classes than to consider multiclass in a problem. This is because the decision boundaries in the first case can be simpler. The first part of the experiments in this chapter is using

two classes, where records matching to the normal class are reported as normal data, otherwise are considered as attacks.

However, to deal with a problem having more than two classes, there are two popular techniques: "One-Vs-One" (OVO) and "One-Vs-All" (OVA). Given a classification problem with M classes ($M > 2$), OVO approach on one hand divides an M-class problem into $\frac{M*(M-1)}{2}$ binary problems. Each problem is handled by a separate binary classifier, which is responsible for separating data of a pair of classes. OVA approach, on the other hand, divides an M-class problem into M binary problems. Each problem is handled by a binary classifier, which is responsible for separating data of a single class from all other classes. Obviously, the OVO approach requires more binary classifiers than OVA. Therefore, it is more computationally intensive. Rifkin and Klautau [81] demonstrated that the OVA technique was preferred over OVO. As such, the OVA technique is applied to the proposed IDS to distinguish between normal and abnormal data using the LS-SVM method. Therefore, if the classifier model confirms that the record is abnormal, the subclass of the abnormal record (type of attacks) can be used to determine the record's type. Algorithm 3 and Algorithm 4 describe the detection processes.

Algorithm 3 Intrusion detection based on LS-SVM {Distinguishing intrusive network traffic from normal network traffic in the case of multiclass}

Input: LS-SVM Normal Classifier, selected features (normal class), an observed data item x

Output: L_x - the classification label of x

begin
$L_x \leftarrow$ classification of x with LS-SVM of Normal class
if $L_x ==$ "*Normal*" **then**
  Return L_X
else
  do: Run Algorithm 4 to determine the class of attack
end
end

Algorithm 4 Attack classification based on LS-SVM

Input: LS-SVM Normal Classifier, selected features (normal class), an observed data item x

Output: L_x - the classification label of x

begin

$L_x \leftarrow$ classification of x with LS-SVM of DoS class

if L_x=="*DoS*" **then**
  Return L_X
else
  $L_x \leftarrow$ classification of x with LS-SVM of Probe class
  if $L_x ==$ "*Probe*" **then**
    Return L_X
  else
    $L_x \leftarrow$ classification of x with LS-SVM of R2L class
    if $L_x ==$ "*R2L*" **then**
      Return L_X
    else
      $L_x ==$ "U2R";
      Return L_X
    end
  end

4.3 Experimental Results and Analysis

4.3.1 Experimental Setup

In all experiments, the value of MI is estimated using the estimator proposed by Kraskov et al. [56] (discussed in Appendix B). To select the best value of k used in the estimator for the approach of k-nearest neoghbors, several experiments with different values for k are conducted. Through the experiments, we have found that the best estimated value of MI was achieved when $k = 6$, which is the same as the value suggested in [56]. In addition, the control parameter β for MIFS algorithm is varied in the range of [0,1], which is the range suggested in [57] and [58], with a step size of 0.1. The optimal value of β that gives the best accuracy rate is selected for a comparison with the proposed approach.

Empirical evidence shows that 0.3 is the best value for β in the three datasets, so we included the results with this optimal β value for comparison. We have also included the results with the value of β equal to 1, which is the same as the value applied in [58]. The reason of choosing different values of β is to test all possibilities of the feature rankings since the best value is undefined for the given problem. The experimental results of different values of β indicate that when the value is closer to 1 the MIFS algorithm assigns larger weights to the redundant features. In other words, the algorithm places more emphasis on the relation between input features rather than between input features and the class and vice versa.

Based on the above findings, to demonstrate the superiority of the proposed feature selection algorithm, five LSSVM-IDSs are built based on all features and the features that are chosen using four different feature selection algorithms (i.e., the proposed

FMIFS, MIFS ($\beta = 0.3$), MIFS ($\beta = 1$), FLCFS), respectively, with $k = 6$. Three different datasets, namely KDD Cup 99 [82], NSL-KDD [71] and Kyoto 2006+ dataset [72], are used to evaluate the performance of these IDSs . The experimental results of the LSSVM-IDS based on FMIFS are compared with the results using the other four LSSVM-IDSs and several other state-of-the-art IDSs.

For the experiments on Kyoto 2006+ dataset, the data of 27, 28, 29, 30 and 31 August 2009 are selected, which contain the latest updated data. For the experimental aims on each dataset, 152460 samples are randomly selected. A 10-fold cross-validation is used to evaluate the detection performance of the proposed LSSVM-IDS. In addition, in order to make a comparison with the detection system proposed in [83], the same sets of data captured from 1st to 3rd November 2007 are chosen for evaluation too. The comparison results are shown in Table 4.6.

4.3.2 Performance Evaluation

Several experiments have been conducted to evaluate the performance and effectiveness of the proposed LSSVM-IDS. For this purpose, the accuracy and F-measure metrics are applied. The accuracy metric is given by

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}, \tag{4.6}$$

where

- True Positive (TP) is the number of actual attacks classified as attacks,

- True Negative (TN) is the number of actual normal records classified as normal ones,
- False Positive (FP) is the number of actual normal records classified as attacks, and
- False Negative (FN) is the number of actual attacks classified as normal records.

The F-measure is a harmonic mean between precision p and recall r [84]. In other words, it is a statistical technique for examining the accuracy of a system by considering both precision and recall of the system. F-measure is given by Equation (4.7)

$$F - measure = \frac{(\beta^2 + 1)(Precision * Recall)}{\beta^2 * Precision + Recall}, \beta = 1. \tag{4.7}$$

The precision is the proportion of predicted positives values which are actually positive. The precision value directly affects the performance of the system. A higher value of precision means a lower false positive rate and vice versa. The precision is given by Equation (4.8).

$$Precision = \frac{TP}{TP + FP}. \tag{4.8}$$

The recall is another important value for measuring the performance of the detection system and to indicate the proportion of the actual number of positives which are correctly identified. The recall is defined as:

TABLE 4.1: Comparison of feature ranking

(A) Feature ranking results on the KDD Cup 99 dataset

Algorithm	# Feature	Feature ranking
FMIFS	19	f_5, f_{23}, f_6, f_3, f_{36}, f_{12}, f_{24}, f_{37}, f_2,f_{32}, f_9, f_{31}, f_{29}, f_{26}, f_{17}, f_{33}, f_{35}, f_{39}, f_{34}
MIFS (β=0.3)	25	f_5, f_{23}, f_6, f_9, f_{32}, f_{18}, f_{19}, f_{15}, f_{17}, f_{16}, f_{14}, f_7, f_{20}, f_{11}, f_{21}, f_{13}, f_8, f_{22}, f_{29}, f_{31}, f_{41}, f_1, f_{26}, f_{10}, f_{37}
MIFS (β=1)	25	f_5, f_7, f_{17}, f_{32}, f_{18}, f_{20}, f_9, f_{15}, f_{14}, f_{21}, f_{16}, f_8, f_{22}, f_{19}, f_{13}, f_{11}, f_{29}, f_1, f_{41}, f_{31}, f_{10}, f_{27}, f_{26}, f_{12}, f_{28}
FLCFS	17	f_{23}, f_{29}, f_{12}, f_{24}, f_3, f_{36}, f_{32}, f_2, f_8, f_{31}, f_{25}, f_1, f_{11}, f_{39}, f_{10}, f_4, f_{19}

(B) Feature ranking results on the NSL-KDD dataset

Algorithm	# Features	Feature ranking
FMIFS	18	f_5, f_{30}, f_6, f_3, f_4, f_{29}, f_{12}, f_{33}, f_{26}, f_{37},f_{39}, f_{34}, f_{25}, f_{38}, f_{23}, f_{35}, f_{36}, f_{28}
MIFS (β=0.3)	23	f_5, f_3, f_{26}, f_9, f_{18}, f_{22}, f_{20}, f_{21}, f_{14}, f_8, f_{11}, f_{12}, f_7, f_{17}, f_{16}, f_{19}, f_1, f_{15}, f_{41},f_{32}, f_{13}, f_{28}, f_{36}
MIFS (β=1)	28	f_5, f_{22}, f_9, f_{26}, f_{18}, f_{20}, f_{14}, f_{21}, f_{16}, f_8, f_{11}, f_1, f_{17}, f_7, f_{12}, f_{19}, f_{15}, f_{40}, f_{32}, f_{13}, f_{10}, f_{28}, f_{31}, f_{27}, f_2, f_{36}, f_{23}, f_3
FLCFS	22	f_{29}, f_{12}, f_{33}, f_{39}, f_4, f_{23}, f_{34}, f_{25}, f_{26}, f_{38}, f_8, f_{35}, f_{19}, f_{32}, f_{18}, f_3, f_6, f_{40}, f_{30}, f_5, f_{27}, f_{22}

(C) Feature ranking results on the Kyoto 2006+ dataset

Algorithm	# Feature	Feature ranking
FMIFS	4	f_{19}, f_{10}, f_2, f_4
MIFS (β=0.3)	6	f_{19}, f_2, f_{10}, f_{16}, f_7, f_{12}
MIFS (β=1)	15	f_{19}, f_7, f_{16}, f_6, f_{12}, f_{11}, f_{17}, f_{13},f_8, f_{15}, f_{18}, f_5, f_9, f_1, f_2
FLCFS	7	f_{10}, f_{17}, f_2, f_{12}, f_8, f_6, f_5

$$Recall = \frac{TP}{TP + FN}. \tag{4.9}$$

4.3.3 Results and Discussion

The classification performance of the detection model combined with FMIFS, MIFS ($\beta = 0.3$), MIFS ($\beta = 1$), FLCFS and all features based on the three datasets are shown in Table 4.2 and Figure 4.2. The results clearly demonstrate that the classification performance of an IDS is enhanced by the feature selection step. In addition, the proposed feature selection algorithm FMIFS shows promising results in terms of low computational cost and high classification results.

Table 4.2 summarises the classification results of the different selection methods in regard to detection rates, false positive rates and accuracy rates. It shows clearly that the detection model combined with the FMIFS achieved an accuracy rate of 99.79%, 99.91% and 99.77% for KDD Cup 99, NSL-KDD and Kyoto 2006+, respectively, and so significantly outperforms all other methods. In addition, the proposed detection model enjoys the highest detection rate and the lowest false positive rate in comparison with other combined detection models.

The proposed feature selection algorithm is computationally efficient when it is applied to the LSSVM-IDS. Figure 4.2 shows the building (training) and test times consumed by the detection model using FMIFS compared with the detection model using all features. The figure shows that the LSSVM-IDS + FMIFS performs better than using all features in all datasets. There are significant differences when performing experiments on KDD Cup 99 and NSL-KDD and a slight difference on Kyoto 2006+ dataset by comparison with the two aforementioned models.

TABLE 4.2: Performance classification for all attacks based on the three datasets

	KDD Cup 99			NSL-KDD			Kyoto 2006+		
	DR	*FPR*	*Accuracy*	*DR*	*FPR*	*Accuracy*	*DR*	*FPR*	*Accuracy*
LSSVM-IDS + FMIFS	**99.46**	**0.13**	**99.79**	**98.76**	**0.28**	**99.91**	**99.64**	**0.13**	**99.77**
LSSVM-IDS + MIFS (β=0.3)	99.38	0.23	99.70	95.96	0.53	97.96	98.59	0.16	99.32
LSSVM-IDS + MIFS (β=1)	89.26	0.34	97.63	93.26	0.47	96.75	98.10	0.58	99.12
LSSVM-IDS + FLCFS	98.47	0.61	98.41	92.29	0.41	96.45	98.07	0.82	98.99
LSSVM-IDS + All features	99.16	0.97	99.19	91.12	0.38	95.96	94.29	0.33	97.42

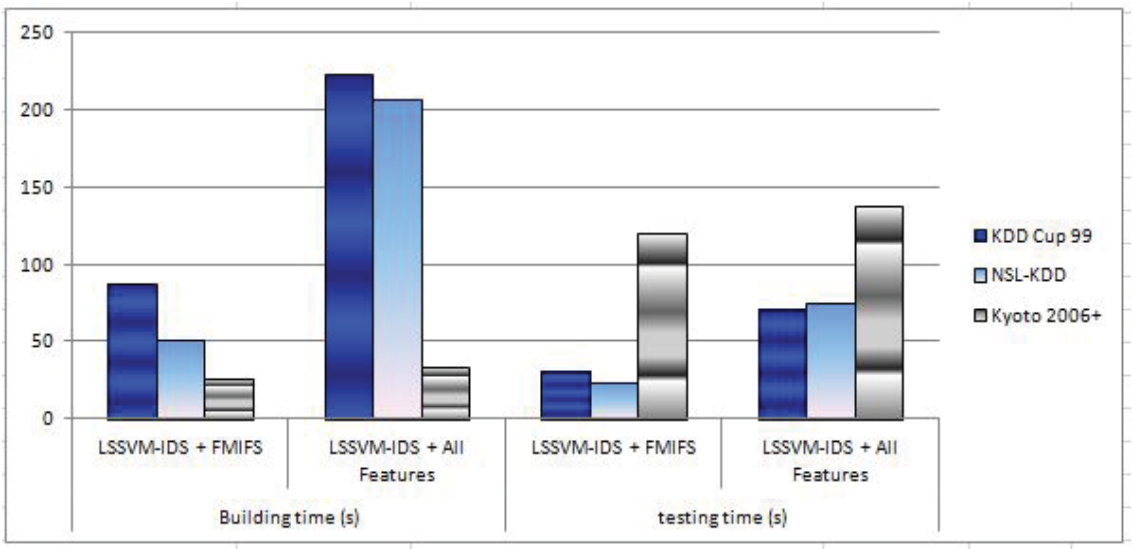

FIGURE 4.2: Building and testing time using all features and FMIFS, respectively, on three datasets.

4.3.4 Comparative Study

In order to demonstrate the performance of the LSSVM-IDS + FMIFS, experiments have been conducted to make comparisons with some state-of-the-art approaches. As mentioned in Section 4.2, the KDD Cup 99 is divided into five different classes and many experiments have been conducted on DoS, Probe, U2R and R2L attacks. Table 4.3 shows the selected features for the different attack classes. Table 4.4, Table 4.5 and Table 4.6 depict the comparison results based on KDD Cup test, NSL-KDDTrain+ and Kyoto 2006+ datasets respectively. The results illustrated in these tables strongly indicate that the proposed detection model shows promising results compared with other models.

Table 4.4 shows the accuracy percentage achieved by different detection models for the five classes on KDD Cup 99 dataset. Regarding the results obtained by other authors, it can be seen that the proposed approach enjoys the best accuracy among all models in all of the classes.

TABLE 4.3: Feature ranking results for the four types of attacks on the KDD Cup 99 dataset

Class	# Feature	Feature ranking
DoS	12	f_{23}, f_5, f_3, f_6, f_{32}, f_{24}, f_{12}, f_2, f_{37}, f_{36}, f_8, f_{31}
Probe	19	f_5, f_{27}, f_3, f_{35}, f_{40}, f_{37}, f_{33}, f_{17}, f_{41}, f_{30}, f_{34}, f_{28}, f_{22}, f_4, f_{24}, f_{25}, f_{19}, f_{32}, f_{29}
U2R	23	f_{37}, f_{17}, f_8, f_{18}, f_{16}, f_1, f_4, f_{15}, f_7, f_{22}, f_{20}, f_{21}, f_{31}, f_{19}, f_{12}, f_{13}, f_{14}, f_6, f_{32}, f_{29}, f_3, f_{40}, f_2
R2L	15	f_3, f_{15}, f_5, f_{10}, f_9, f_{32}, f_{33}, f_{22}, f_1, f_{17}, f_{24}, f_{11}, f_{23}, f_8, f_6

Table 4.5 demonstrates the result achieved by LSSVM-IDS + FMIFS compared with other approaches tested on NSL-KDDTrain+ datasets in terms of the detection, false positive and accuracy rate. It is clear that LSSVM-IDS + FMIFS enjoys the best results at 99.94% accuracy, 98.93% detection rate and 0.28% false positive rate.

Table 4.6 shows a comparison with the results achieved by CSV-ISVM proposed in [83] that has been tested on Kyoto 2006+ dataset. Through the results, both systems show continuous improvement in detection rates and reduction in false positive rates. However, from the very first iteration, the obtained results of the LSSVM-IDS are better, compared to CSV-ISVM. The final results achieved by LSSVM-IDS in the 10th iteration show 97.80% and 0.43% of the final detection and false positive rates respectively, while CSV-ISVM produces 90.15% and 2.31% of the final detection and false positive rates respectively. The training and testing

TABLE 4.4: Comparison results in terms of accuracy rate with other approaches based on the KDD Cup 99 dataset (n/a means not available by authors)

System	Normal	DoS	Probe	U2R	R2L
LSSVM-IDS + FMIFS	**99.75**	**99.86**	**99.91**	**99.97**	**99.92**
SVM with PBR [85]	99.59	99.22	99.38	99.87	99.78
SVM [25]	99.55	99.25	99.70	99.87	99.78
Bayesian Network [86]	98.78	98.95	99.57	48.00	98.93
Flexible Neural Tree [87]	99.19	98.75	98.39	99.70	99.09
SVM + PSO and FS [88]	99.45	n/a	n/a	n/a	n/a
SVM + SA and FS [89]	99.42	n/a	n/a	n/a	n/a
TUIDS [90]	94.76	n/a	n/a	n/a	n/a
Radial SVM [27]	n/a	98.94	97.11	97.80	97.78

TABLE 4.5: Comparison results based on NSL-KDD dataset (n/a means not available by authors)

System	# Feature	DR	FPR	Accuracy
LSSVM-IDS + FMIFS	18	**98.93**	**0.28**	**99.94**
DMNB [76]	all	n/a	3.0	96.50
DBN-SVM [91]	all	n/a	n/a	92.84
Bi-layer behavioral-based [92]	20	n/a	n/a	99.20
TUIDS [90]	all	98.88	1.12	96.55
FVBRM [93]	24	n/a	n/a	97.78
C4.5 with linear correlation-based [94]	17	n/a	n/a	99.10
PSOM [75]	10	n/a	n/a	88.30
HTTP based IDS [95]	13	99.03	1.0	99.38
Hybrid IDS [96]	all	99.10	1.2	n/a

times taken by both systems are also demonstrated in Table 4.6. Unlike CSV-ISVM, LSSVM-IDS take much less time. This is because LSSVM-IDS was using a feature selection stage that reduced the number of needed features for the classifier to five features. These features are: {*source_IP_address, service, dst_host_srv_count, destination bytes, src_bytes*}.

TABLE 4.6: Comparison performance of classification on the Kyoto 2006+ dataset (the days 2007, Nov. 1,2 and 3), #I is the number of Iteration

#I	LSSVM-IDS + FMIFS				CSV-ISVM [83]			
	DR	FPR	$Train(s)$	$Test(s)$	DR	FPR	$Train(s)$	$Test(s)$
1	96.01	0.84	0.152	0.246	79.65	4.54	1.823	7.76
2	97.01	0.64	0.296	0.396	84.72	4.03	3.463	10.363
3	97.13	0.64	0.505	0.656	85.58	3.92	5.26	15.443
4	97.18	0.64	1.140	1.343	86.08	3.80	9.662	19.532
5	97.26	0.60	1.475	1.773	86.81	3.54	11.302	22.735
6	97.32	0.57	2.228	2.643	87.24	3.33	13.593	25.887
7	97.61	0.55	3.214	3.773	88.08	3.03	14.348	28.23
8	97.61	0.53	4.343	5.172	88.10	3.01	17.475	31.615
9	97.70	0.45	5.585	6.508	89.64	2.52	23.02	35.547
10	97.80	0.43	7.275	8.408	90.15	2.31	27.257	40.097

4.3.5 Additional Comparison

The performance of the LSSVM-IDS model was further compared with the PLSSVM model [1], which also used a feature selection algorithm based on the mutual information method, named MMIFS. The comparison results shown in Table 4.7 are based on the corrected labels dataset. The effectiveness of the two models is compared in three aspects: the accuracy rate, average building time and testing time in minutes.

From Table 4.7, it can be observed that the proposed system reduces the building time and testing time very considerably for all categories. In addition, with respect to the accuracy both models have shown promising results for all classes. It is clear

TABLE 4.7: Accuracy, building time (min) and testing time (min) for all different classes on corrected labels of KDD Cup 99 dataset compare with PLSSVM proposed by Amiri in [1].

Class Name	Model	Accuracy (%)	Building time (min)	Testing time (min)
Normal	LSSVM-IDS + FMIFS	98.39	**7.92**	**5.51**
	PLSSVM + MMIFS	**99.1**	25	11
DoS	LSSVM-IDS + FMIFS	**98.93**	**10.06**	**4.50**
	PLSSVM + MMIFS	84.11	19	8
Probe	LSSVM-IDS + FMIFS	**99.57**	**13.04**	**8.49**
	PLSSVM + MMIFS	86.12	35	13
U2R	LSSVM-IDS + FMIFS	**99.66**	**0.47**	**0.32**
	PLSSVM + MMIFS	99.47	23	10
R2L	LSSVM-IDS + FMIFS	90.08	**1.06**	**0.44**
	PLSSVM + MMIFS	**98.70**	5	4
Overall	LSSVM-IDS + FMIFS	**97.33**	**6.51**	**3.85**
	PLSSVM + MMIFS	93.50	21.4	9.20

from the table that LSSVM-IDS has better accuracy in DoS, Probe and U2R classes, while the PLSVM produces a better accuracy rate when applied to Normal and R2L class. Moreover, the table shows that LSSVM-IDS outperforms the PLSSVM model in the overall performance.

Furthermore, the detection rate of LSSVM-IDS has been compared and shown in Table 4.8 with some other approaches that have been tested on the corrected labels dataset. Through Table 4.8, compared to the KDD Cup 99 winner's detection system and other systems, LSSVM-IDS achieves the best detection rates for U2R and R2L attacks with rates of 22.11% and 88.38% respectively. The detection model proposed in [70] provides the best detection rate for the Probe attack of 97.5%. For the normal class, all of the KDD Cup 99 winner [66], Association rule [69] and

TABLE 4.8: Detection rate (%) for different algorithm performances on the test dataset with corrected labels of KDD Cup 99 dataset (n/a means not available by authors)

System	Normal	DoS	Probe	U2R	R2L	Overall
LSSVM-IDS + FMIFS	98.98	98.76	86.08	**22.11**	**88.38**	**78.86**
KDD'99 winner [66]	**99.50**	97.10	83.30	13.20	8.40	60.3
Kernel Miner [67]	99.42	97.47	84.52	11.84	7.32	60.11
PNrule [97]	**99.50**	96.90	73.20	6.60	10.70	57.38
SVM IDS [68]	99.3	91.6	36.65	12	22	52.31
Association rule [69]	**99.50**	96.80	74.90	3.8	7.9	56.58
ESC-IDS [28]	98.20	**99.50**	84.10	14.10	31.50	65.48
Clustering [70]	99.3	99.5	**97.5**	19.7	28.8	68.96
TUIDS [90]	90.01	n/a	n/a	n/a	n/a	n/a

PNrule [97] achieve the best result with 99.50% detection rate. However, overall LSSVM-IDS has achieved the best detection rate among all systems.

Figure 4.3 illustrates a comparison between LSSVM-IDS and the other two detection models proposed by Tsang [98] in terms of F-measure rates. These two methods have applied the genetic-fuzzy rule mining technique to evaluate the importance of IDS features. This figure, makes it obvious that the proposed model outperforms Tsang models in most of the classes including Normal, DoS, Probe and R2L with 89.31%, 99.27%, 84.16 and 48.13%, respectively. MOGFIDS provides the highest result in U2R class of 25.09%. Overall, the results of the LSSVM-IDS shown in this figure demonstrate satisfying performance improvements compared with the other two methods.

Figure 4.4 shows a comparison between those system proposed in [71], [99] and [100] that have been tested on the KDDTest^{-}21 in terms of the classification accuracy.

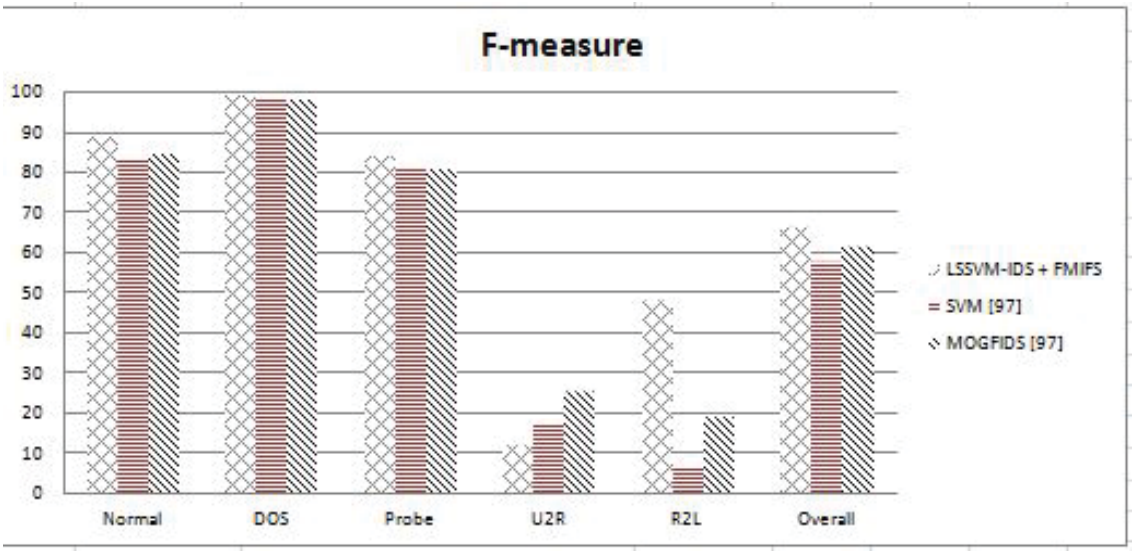

FIGURE 4.3: Comparison results of $F-$measure rate on the corrected labels of KDD Cup 99 dataset

Among those systems, the proposed detection model achieved the best classification accuracy of 94.68%.

To sum up, the large number of unseen attacks in these datasets that do not appear in the training datasets make it even harder for an IDS. For example in the corrected labels dataset, Bouzida [101] has shown that snmpgetattck and normal records have almost the same features, and this makes it impossible for any IDS to detect this type of attack.

4.4 Summary

Recent studies have shown that two main components are essential to build an IDS. They are a robust classification method and an efficient feature selection algorithm.

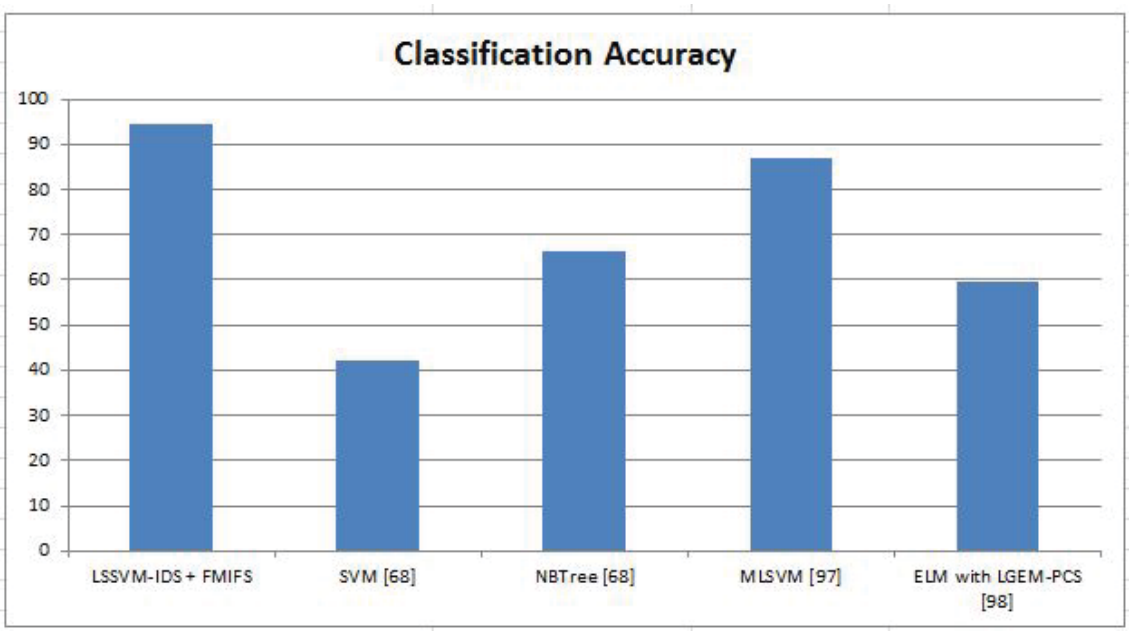

FIGURE 4.4: Comparison results of classification accuracy on KDDTest^{-}21

In this chapter, a supervised filter-based feature selection algorithm has been proposed, namely Flexible Mutual Information Feature Selection (FMIFS). FMIFS is an improvement over MIFS and MMIFS. FMIFS suggests a modification to Battiti's algorithm to reduce the redundancy among features. FMIFS eliminates the redundancy parameter β required in MIFS and MMIFS. This is desirable in practice since there is no specific procedure or guideline to select the best value for this parameter.

FMIFS is then combined with the LSSVM method to build an IDS. LSSVM is a least square version of SVM that works with equality constraints instead of inequality constraints in the formulation designed to solve a set of linear equations for classification problems rather than a quadratic programming problem. The proposed LSSVM-IDS has been evaluated using three well known intrusion detection

datasets: KDD Cup 99, NSL-KDD and Kyoto 2006+ datasets. The performance of LSSVM-IDS using kddcup testdata, KDDTest+ and the days 2007, Nov. 1,2 and 3 of Kyoto dataset has exhibited better classification performance in terms of classification accuracy, detection rate, false positive rate and F-measure than some of the existing detection approaches. In addition, the proposed LSSVM-IDS has shown comparable results with other state-of-the-art approaches when using the corrected labels KDD Cup 99 dataset and tested on Normal, DoS, and Probe classes; it outperforms other detection models when tested on U2R and R2L classes. Furthermore, for the experiments on the KDDTest^{-}21 dataset, LSSVM-IDS produces the best classification accuracy among other detection systems tested on the same dataset. Finally, based on the experimental results achieved on all datasets, it can be concluded that the proposed detection system has achieved promising performance in detecting intrusions over computer networks. Overall, LSSVM-IDS has performed the best when compared with the other state-of-the-art models.

However, the detection performance of the proposed intrusion detection system needs to be further enhanced in order to achieve a high level of defence and strengthen network security against malicious attacks. There are a number of research directions that can be use to extend the work achieved in this chapter. Thus, the following chapter puts the classification performance as the main target and proposes a hybrid feature selection algorithm in combination with filter and wrapper methods.

Chapter 5

Supervised Hybrid Feature Selection Algorithm for IDS

As discussed in Chapter 2, the filter-based feature selection methods have less computational cost and are easy to apply. However, the lack of interaction between the classifier and the dependence among features make these methods fail to select the optimal available subset [47]. The wrapper methods are considered to be more accurate but often have much more computational complexity when dealing with a large volume of data and high dimensional data compared to filter approaches [48]. The hybrid methods, on the other hand, exploit the advantages of both filter and wrapper methods. These methods utilise both an independent measure and a fitness evaluation function of the feature subset to select the final optimal subset of features [49]. Hybrid methods are considered to be more effective and can achieve promising classification performance.

In this chapter, a hybrid feature selection algorithm is designed. The proposed approach is a combination of two main stages: (1) filter feature ranking; and (2) wrapper-based Improved Forward Floating Selection (IFFS) using LSSVM and classification accuracy. The filter method aims to reduce the computational cost of the wrapper search by eliminating irrelevant and redundancy features from the initial feature set. This phase applies the filter method FMIFS proposed in Chapter 4. The wrapper method is used to search for a proper subset that improves the classification accuracy. The aim of the proposed hybrid method is to achieve both the high accuracy of wrapper approaches and the efficiency of filter approaches. Finally, in order to examine the effectiveness and feasibility of the proposed feature selection method, the final proper subset is then passed through LSSVM classifier to build an IDS.

The outline of this chapter is as follows. Section 5.1 describes the principle of the improved forward floating selection algorithm. Section 5.2 introduces the proposed hybrid feature selection algorithm. Section 5.3 details the detection framework showing the different detection stages. Section 5.4 presents the experimental details and results. Finally, a summary of the chapter is drawn in Section 5.5.

5.1 Improved Forward Floating Selection

The sequential search looks for the optimal feature subset by either adding (or removing) one feature at a time until the specified criteria is reached. Sequential Forward/Backward Selection (SFS/SBS) are two of the most commonly used searching techniques in selecting the most optimal subsets and decreasing very large

feature sets [102]. SFS starts with an empty set and incrementally adds features to the selected subset based on their importance. SBS, on the other hand, starts with all features and deletes one feature at a time. However, these methods suffer from the so called "nesting effect" problem. Once a feature is added (or deleted), it will not be considered in upcoming selection iterations.

Sequential Forward/Backward Floating Search (SFFS/ SBFS) have been successfully applied to overcome the "nesting effect" problem by backtracking after each sequential iteration to select a better subset [103]. The SFFS method starts the search with an empty set and uses the SFS to add one feature at a time to the selected feature set. Every time a new feature is added, the SFFS algorithm uses SBS to backtrack and remove one feature at a time to find a better subset. The SBFS method starts the search from all input features and uses the SBS to remove one feature at a time from the original feature set. After each backward step, SFBS performs forward steps to find a better subset that can produce a better performance.

Improved Forward Floating Selection (IFFS) [104] was introduced to improve the selection process in the SFFS algorithms. The IFFS adds an additional search step together with the backtracking step called "replace weak features". The method further investigates the feature subset if removing an old feature and adding a new one to a selected subset at each iteration will improve the quality of the selected subset.

5.2 Proposed Hybrid Feature Selection

This section proposes a hybrid feature selection approach that combines the advantages of both filter and wrapper methods. The framework of the proposed algorithm, as shown in Figure 5.1, consists of two main phases. The first is the upper phase at which the mutual information is used for feature ranking and elimination. The second is the lower phase which determines the optimal subset of features S_{best} and contributes the maximum classification accuracy on the training dataset.

Suppose that the total number of features considered in the dataset is n. The filtering process is applied to rank the features incrementally with a starting value at 1 to eliminate any irrelevant and redundant features from the initial features set. This phase will be continued until L features are selected. The wrapper method is applied to evaluate all possible sets to select the best feature set that produces the best classification accuracy among other available subsets.

5.2.1 Filter Method for Feature Pre-selection

The Flexible Mutual Information Feature Selection algorithm (FMIFS) proposed in the previous chapter is applied in the upper phase of the proposed hybrid method. It is designed to eliminate irrelevant and redundancy features from the original data. This helps the wrapper method (the lower phase) to decrease the searching range from the entire original feature space to the pre-selected features (the output of the upper phase).

As discussed in Chapter 4, FMIFS algorithm searches for relevant features by looking at the characteristics of each individual feature using mutual information as

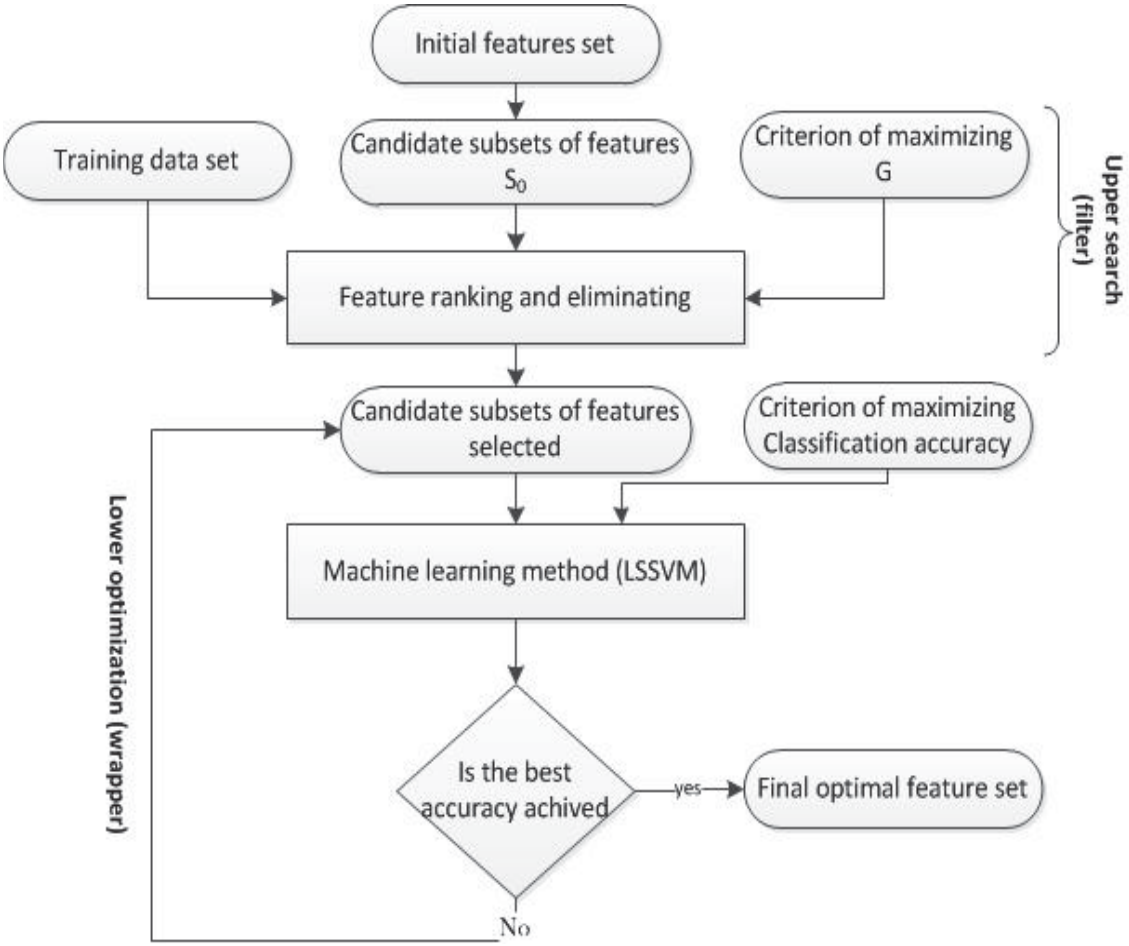

FIGURE 5.1: The overall scheme of the proposed hybrid feature selection

an evaluation criterion to guide the selection process. It selects the feature that maximises the term $I(C; f_i)$, which represents the amount of information that feature f_i carries about the class C, corrected by subtracting the average Minimum Redundancy (MR) between the candidate feature and the set of previously selected features. Therefore, FMIFS intends to determine the feature that maximises the term G in Equation (5.1).

$$G = I(C; f_i) - \frac{1}{\mid S \mid} \sum_{f_s \in S} MR, \tag{5.1}$$

where MR is the relative minimum redundancy of feature f_i against feature f_s and is denoted by

$$MR = \frac{I(f_i; f_s)}{I(C; f_i)}. \tag{5.2}$$

5.2.2 Wrapper-based IFFS for Feature Selection Using LS-SVM

Once the filter method finishes its task, the lower phase evaluates all possible subsets that can be generated from the output subset of the upper phase in a wrapper manner. This is to determine the optimal subset of feature S_{best} that can produce the best classification performance. To do so, LS-SVM and the classification accuracy are employed. If the performance reaches the best accuracy rate, the selection process is completed and the output is the last optimal subset of features with cardinality of ω. Otherwise, the selection procedure carries the searching at cardinality of m+1 by adding one feature from the remaining features, replacing the weak features that produce low accuracy and repeating the above steps. Figure 5.2 shows the overall scheme of the wrapper-based IFFS.

As shown in Figure 5.2, the wrapper phase involves two important steps: (1) backtracking and (2) replacing the weak feature.

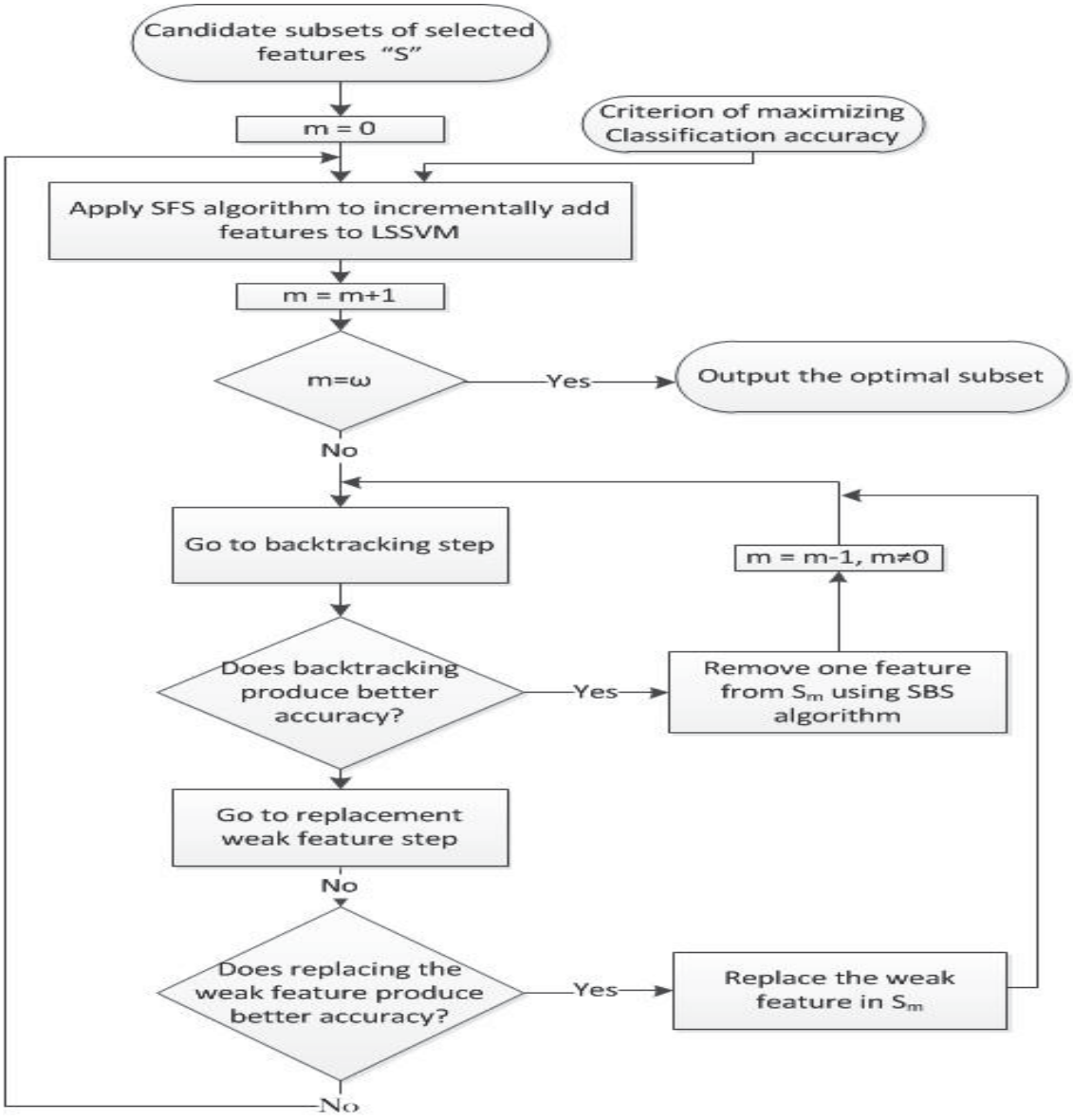

FIGURE 5.2: The overall procedure of the proposed wrapper algorithm-based IFFS

5.2.2.1 Backtracking

To avoid the "nesting problem" [48], the proposed algorithm uses SFS to add one feature at a time to the subset of features. When a new feature is added to the

current selected subset, the algorithm uses SBS for backtracking to remove one feature in each iteration to find a better subset.

5.2.2.2 Replacing the Weak Feature

The proposed algorithm not only backtracks to find the best subset but also attempts to find if replacing weak features in the current selected feature set can provide a better subset. The aim is to further investigate if removing one feature in the selected feature set and adding a new one using SFS can enhance the classification accuracy of the current selected feature set.

5.3 Intrusion Detection Framework Based on LS-SVM

The framework of the proposed IDS, as shown in Figure 5.3, is comprised of the same four stages as discussed in Chapter 3 and Chapter 4. These stages are: (A) data collection, where a sequence of network packets is collected; (B) data preprocessing, where training and test data are preprocessed; (C) classifier training, where the training data is trained for the classification problem; and (D) attack recognition, where the classifier is trained using LS-SVM to detect intrusions on the test data.

Compared to the framework proposed in Chapter 4, this chapter employs the proposed hybrid feature selection. The output of the proposed feature selection is then used to build the classification model of the detection system. More details about the framework stages can be found in Chapter 3 and Chapter 4.

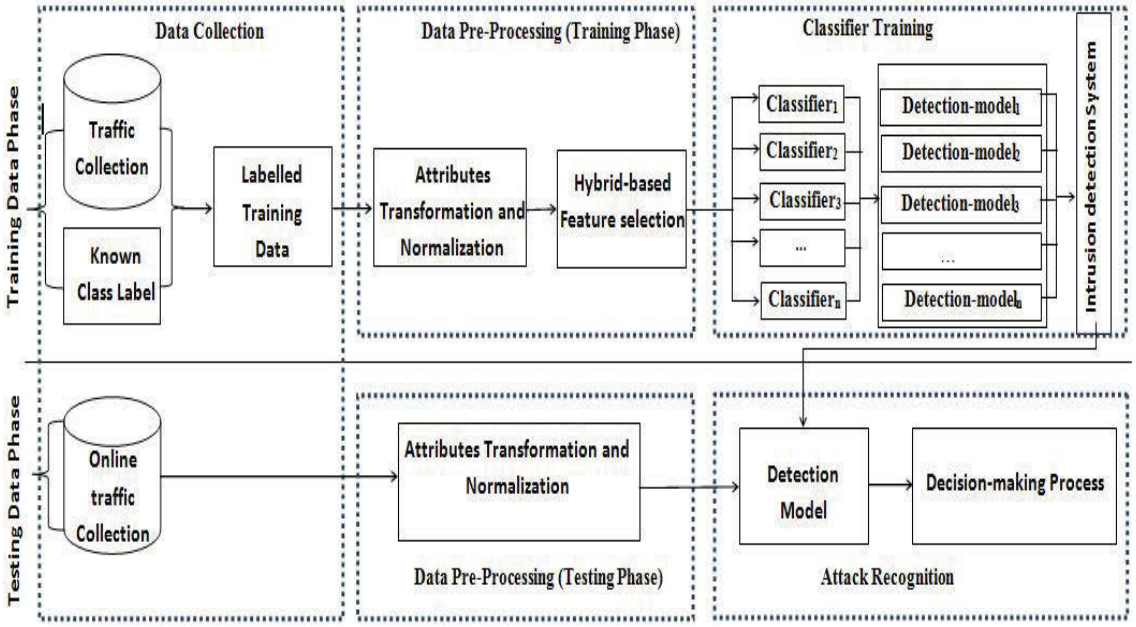

FIGURE 5.3: The framework of the LS-SVM-based intrusion detection system

Algorithm 5 The detection algorithm to distinguish intrusive network traffic from normal network traffic

Input: LS-SVM Normal Classifier, selected features (normal class), an observed data item x

Output: L_x-the classification label of x

begin $L_x \leftarrow$ classification of x with LS-SVM of Normal class

if $L_x =$ "*Normal*" **then**

| Return L_X

else

| $L_x =$ "Abnormal"

| Return L_X

end

After completing the whole iteration process, the final optimal subset of features is determined which includes the most correlated features for the class and can differentiate between the normal and intrusion traffics using the saved trained model. The testing data is then passed through the trained model to detect intrusions. As shown in algorithm 5, records matching the normal class are considered as normal

data, otherwise they are reported as attacks.

5.4 Experiments and Results

To facilitates a fair and rational comparison with other previously proposed detection approaches, the KDD Cup 99 dataset and Kyoto 2006+ dataset are utilised to evaluate the performance of the proposed detection system. As shown in the literature review, a significant number of state-of-the-art IDSs, such as those in [25, 27, 85–87] , were evaluated using "10% KDD Cup 99" data. Therefore, training and testing the proposed detection system on the "10% KDD Cup 99" data can assist to provide a fair comparison with those systems. As discussed in Chapter 2, the "10% KDD Cup 99" contains about 494,021 TCP/IP connection records. Such large size data cannot be fed to an LS-SVM classifier in the training phase, so 15,246 records from the two different classes are randomly selected as the training data and the remaining 478,775 (= 494,021 - 15,246) samples are used for evaluation purposes. Both the training and testing samples used in these experiments consist of 41 features.

Furthermore, to validate the performance of the hybrid feature selection, the proposed detection model is tested using the corrected labels KDD Cup 99. This dataset has been used to validate some of state-of-the-art IDSs such as those in [28, 66–68, 70]. Therefore, for a fair comparison with those detection systems, this dataset is utilised to test the performance of the detection model. The corrected labels KDD Cup 99 dataset contains 311,029 TCP/IP connection records,

TABLE 5.1: Comparison of feature ranking

(A) Feature ranking results on the KDD Cup 99 dataset

Algorithm	# Feature	Feature ranking
Proposed method	6	f_5, f_3, f_{23}, f_{32}, f_{34}, f_{35}
Filter method	19	f_5, f_{23}, f_6, f_3, f_{36}, f_{12}, f_{24}, f_{37}, f_2,f_{32}, f_9, f_{31}, f_{29}, f_{26}, f_{17}, f_{33}, f_{35}, f_{39}, f_{34}
MIFS (β=0.3)	25	f_5 ,f_{23} ,f_6 ,f_9 ,f_{32} ,f_{18} ,f_{19} , f_{15} ,f_{17} ,f_{16} ,f_{14} ,f_7 ,f_{20} ,f_{11} ,f_{21} ,f_{13} ,f_8, f_{22}, f_{29} ,f_{31},f_{41} ,f_1 ,f_{26} ,f_{10} ,f_{37}
MIFS (β=1)	25	f_5 ,f_7 ,f_{17} ,f_{32} ,f_{18} ,f_{20} ,f_9 , f_{15} ,f_{14} ,f_{21} ,f_{16} ,f_8 ,f_{22} ,f_{19} ,f_{13} ,f_{11} ,f_{29}, f_1, f_{41} ,f_{31},f_{10} ,f_{27} ,f_{26} ,f_{12} ,f_{28}

(B) Feature ranking results on the Kyoto 2006+ dataset

Algorithm	# Feature	Feature ranking
Proposed method	6	f_{19}, f_{10}, f_2, f_4, f_{16}, f_9
Filter method	4	f_{19}, f_{10}, f_2, f_4
MIFS (β=0.3)	6	f_{19}, f_2, f_{10}, f_{16}, f_7, f_{12}
MIFS (β=1)	15	f_{19}, f_7, f_{16}, f_6, f_{12}, f_{11}, f_{17}, f_{13},f_8, f_{15}, f_{18}, f_5, f_9, f_1, f_2

where around 80.6% of the samples are attacks and the remaining ones are normal records.

For these experiments on Kyoto 2006+ dataset, the data of the days 2009 August 27, 28, 29, 30 and 31 are selected, which contain the latest updated data. For training purposes, 15,246 samples are randomly selected and the remaining are used as testing data.

5.4.1 Results and Discussion

For the proposed feature selection algorithm, the search terminates when the number of features in the current selected subset reaches ω to allow enough backtracking. For these experiments, the value of ω is chosen to be six. This choice is not critical, but is to avoid high computational time.

To compare with Battiti's MIFS algorithm, several experiments are conducted with different values for β as discussed in the previous chapter. Similarly, the control parameter β is chosen to be between 0.3 to 1. Then, the best value for β that gives the best accuracy rate is selected for a comparison with the proposed approach. Table 5.1[A-B] shows the selected feature subsets of the different feature selection methods on KDD Cup 99 dataset and Kyoto 2006+ dataset.

As discussed in Chapter 4, experiments using different values for β have shown that 0.3 is the best value for β in this dataset. For the same reason discussed in Chapter 4, the value of β is chosen to be equal to 1, which is the same value applied in [58].

In addition, the results of the detection model using the proposed hybrid feature selection algorithm are compared with the detection model when only using the filter algorithm (discussed in Section 5.2.1). Table 5.2 summarises the classification results of the different selection methods in respect to detection rates, false positive rates, accuracy rates and F-measure. Through Table 5.2, it can be seen that the detection model with the proposed hybrid method achieves the highest accuracy rates with 99.90%. In addition, with respect to the false positive rate and detection rate, the proposed approach obtains the best rates among other approaches with 0.07% and 99.93%, respectively.

The F-measure is also applied to examine the level of accuracy of the different classifiers in relation to the *Precision* (P) and *Recall* (R). F-measure is given by (4)

$$F - measure = \frac{(\beta^2 + 1)(P * R)}{\beta^2 * P + R}, \quad \beta = 1. \tag{5.3}$$

It can be observed from the results that feature selection improves the classification performance in comparison with using all features. In general, in terms of the F-measure results for all methods, the proposed detection method with hybrid feature selection enjoys higher rates.

TABLE 5.2: Performance of classification based on the evaluation data on KDD Cup 99

IDS with:	DR	FR	Accuracy	F-measure
Proposed method	**99.93 ± 0.08**	**0.07 ± 0.04**	**99.90 ± 0.03**	**99.53 ± 0.05**
Filter method	99.43 ± 0.08	0.17 ± 0.02	99.75 ± 0.04	99.34 ± 0.03
MIFS (β=0.3)	99.38 ± 0.14	0.23 ± 0.02	99.70 ± 0.3	99.21 ± 0.09
MIFS (β=1)	99.02 ± 0.04	0.30 ± 0.06	99.57 ± 0.3	98.86 ± 0.09
All features	99.86 ± 0.01	0.97 ± 0.05	99.19 ± 0.04	97.89 ± 0.05

Table 5.3 shows the classification results of the different selection methods in regard to detection rates, false positive rates and accuracy rates based on Kyoto 2006+ dataset. From the table, it is clear that the proposed method enjoys better accuracy and lower false positive rate than other methods. In terms of the detection rate the proposed detection method achieved 99.64%, which is similar to the detection rate achieved by the filter method.

Figure 5.4 shows the average training and testing time (in seconds) of the proposed detection model with hybrid feature selection compared with using only the filter

TABLE 5.3: Performance of classification based on Kyoto 2006+ data

Detection model with:	DR	FR	Accuracy
Proposed method	**99.64**	**0.11**	**99.78**
Filter method	**99.64**	0.13	99.77
MIFS (β=0.3)	98.59	0.16	99.32
MIFS (β=1)	98.10	0.58	99.12
All features	94.29	0.33	97.42

method and those using all 41 features. Through Figure 5.4, it can be observed that the detection model with a feature selection phase has less building and testing times than these using all features. In addition, the proposed approach illustrates the best average times of building and testing processes.

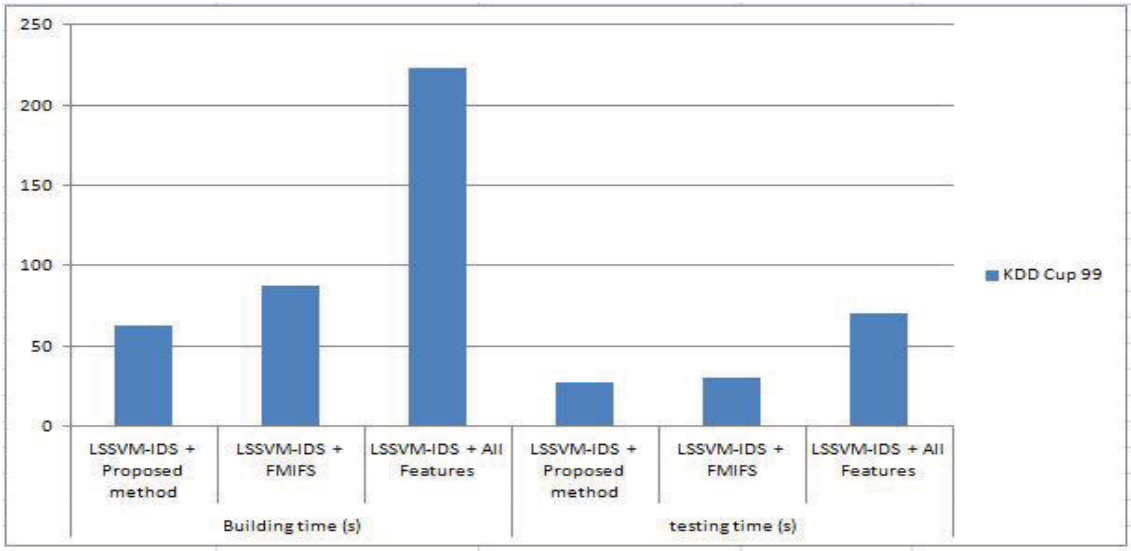

FIGURE 5.4: Building and testing times using all features and FMIFS and the proposed method, respectively, on KDD Cup 99.

To sum up the above tables, the results in all tables strongly indicate that the feature selection algorithm is a necessary step in building an IDS. In addition, compared to

TABLE 5.4: Comparison results in terms of accuracy rate with other approaches based on the evaluation dataset

System	Accuracy rate(%)
Proposed method	**99.90**
Filter method	99.75
SVM with PBR[85]	99.59
Method proposed in [25]	99.55
Bayesian Network [86]	98.78
Flexible Neural Tree [87]	99.19

the filter algorithm and the MIFS methods, the proposed hybrid approach achieves promising classification accuracies, detection rates, false alarm rates and F-measure rates. Furthermore, the proposed method is faster in building and testing times than those methods that need to examine all input features.

5.4.2 Comparative Study

In order to demonstrate the performance of the proposed detection model, several experiments are conducted to compare with some state-of-the-art approaches. Table 5.4 and Table 5.5 depict the comparisons results over the evaluation and test datasets.

Through Table 5.4, the accuracy rate of the proposed detection approach is compared with those approaches that have been evaluated on the "10% KDD Cup 99" dataset. Regarding the results obtained by other authors, it can be seen that the proposed detection approach enjoys the best accuracy over all approaches. Therefore, it can be indicated that the proposed model has shown a good performance in identifying intrusions in network traffic.

TABLE 5.5: Performance of classification based on the corrected labels of KDD Cup 99 data (n/a means not available by authors)

System	DR	FP	Accuracy
Proposed method	99.47	**0.521**	98.90
KDD'99 winner [66]	**99.50**	0.6	91.8
Kernel Miner [67]	99.42	0.6	91.5
Method proposed in [68]	99.3	n/a	n/a
ESC-IDS [28]	98.20	1.9	95.3
Clustering feature [70]	99.3	0.7	95.7
PLSSVM [1]	95.69	0.65	**99.1**

Table 5.5 shows further comparison results with those detection systems that have been evaluated on the corrected labels KDD Cup 99. Approximately 18,729 samples of attacks in this dataset are previously unseen attacks, which only appear in the test dataset and do not appear in the "10% KDD Cup 99". This makes it even harder for an IDS trained by the training dataset to show good accuracy in detecting these attacks.

As shown in Table 5.5, compared to all detection systems, this system score the lowest false positive rate with 0.521%. Although the KDD Cup 99 winner [66] provides better performance than that of this study in terms of detection rates, the difference is insignificant. The PLSSVM [1] shows the best accuracy rate among all systems with 99.1%, while the proposed system achieves the second best with a small difference 98.90%.

5.5 Summary

In this chapter, a hybrid-based feature selection approach in combination with filter and wrapper selection processes has been proposed for feature selection and intrusion detection data classification. The approach features two main phases: (1) filter feature ranking and eliminating phase; and (2) wrapper feature selection using LS-SVM and classification accuracy. The aim is to achieve both the efficiency of filter approaches and the high accuracy of wrapper approaches.

The filter feature ranking is a pre-selection step with the aim of reducing the computational cost of the wrapper search by removing irrelevant and redundancy features from the input feature set. The wrapper method searches for the optimal subset that improves the classification performance by comparing the accuracy of the current selected subset with the previously selected one. This phase employs two main steps: (1) backtracking to avoid the nesting problem and (2) replacing the weak features to check if the replacement of some features can provide a better subset.

The proposed feature selection method has been evaluated through developing an IDS. Two well-known IDS datasets have been used to evaluate the performance of the proposed method. They are KDD Cup 99 data and Kyoto 2006+ data. Two types of KDD Cup 99 datasets have been involved in the evaluation processes of the detection model. Experiments on the "10% KDD Cup 99" dataset exhibit promising results in terms of classification accuracy, low computational cost and F-measure. In addition, compared with those systems that have been evaluated on the corrected labels KDD Cup 99 dataset, the detection model has shown comparable results in terms of detection rate, false positive rate and accuracy rate. Furthermore, the proposed detection method enjoys better performance on Kyoto 2006+ dataset

compared with other comparable methods. Thus, the experimental results achieved on both datasets show that the proposed detection system has achieved a promising performance in detecting intrusions over computer networks.

The proposed feature selection methods in Chapter 4 and Chapter 5 are supervised feature selection methods in which the class labels are required. However, labelled data are not always available, and they are expensive or hard to obtain. Hence, many attempts have been made to develop unsupervised feature selection algorithms that can utilise this data. In Chapter 6 of this thesis, we extend this research to unsupervised feature selection method.

Chapter 6

Unsupervised Feature Selection Algorithm for IDS

Due to the lack of categorised information in many practical applications, unsupervised feature selection has been proven to be more practically important but at the same time more difficult. It is not an easy task to assess the relevance of a feature or a subset of features when there are no labels available with the data. The basic assumption behind unsupervised feature selection techniques is that samples belonging to the same class are probably located close to each other, otherwise they are from a different class.

As shown in Chapter 2, several attempts have been made to develop an intelligent unsupervised feature selection technique which can utilise unlabeled data. The Variance score method is one of the simplest unsupervised feature selection methods that calculates the variance of each of the features individually and selects the ones that have larger variance values [105]. Another unsupervised feature selection

method is the Laplacian score [62]. The Laplacian score algorithm not only selects the features with high variances, but also investigates the locality preserving power of every feature in the data. Unlike the Laplacian score, Local and Global structure preserving (LGFS) [9] not only considers the locality structure preserving power of each feature but also its globality structure preserving. The assumption behind LGFS method is that samples belonging to the same class are probably located close to each other, otherwise they are from a different class. These methods, however, neglect the redundancy among selected features, so they select many redundant features, and affect the classification performance. Ren et al. in [9] proposed an Extended version of LGFS, named Extended Local and Global structure preserving (E-LGFS). E-LGFS applies the normalised mutual information method, that has been proposed in [59], to eliminate redundancy among selected features. However, in many applications (such as many real-world applications), extracting the local structure information is much important than the global structure information in order to find the best features in the data [62, 106].

To address the aforementioned problems on the methods for unsupervised feature selection, this chapter considers the feature selection problem for data classification in the absence of data labels. In this chapter, two unsupervised feature selection algorithms are proposed and they named as Extended Laplacian score EL and Modified Laplacian score ML. These two algorithms are enhanced versions of the Laplacian score method that consider the locality structure preserving power of each feature and the redundancies among features. More specifically, each of EL and ML consists of two main phases. In the first phase, the Laplacian score algorithm is applied to rank the features by evaluating the power of locality preservation for each feature in the initial data. In the second phase, a new redundancy penalization

technique uses mutual information to remove the redundancy among the selected features. This technique makes feature selection in each round of iterations based on the entropies of the already selected features in the previous rounds. ML differs from that of EL in the redundancy measurement. ML measures redundancy between a candidate feature and the previously selected features based on the entropies of all remaining candidate features. The experimental results show that ML performs better than EL and four other state-of-art methods (including the Variance score algorithm and the Laplacian score algorithm) in terms of the classification accuracy.

The outline of this chapter is as follows. Section 6.1 provides a description of the Laplacian Score algorithm. Section 6.2 discusses the proposed unsupervised feature selection method. Section 6.3 details our detection framework showing different detection stages involved in the proposed scheme. Section 6.4 presents the experimental details and results. Finally, a summary to the chapter is drawn in Section 6.5.

6.1 Laplacian Score

To explain the Laplacian Score, we refer to the definition proposed in [62]. Laplacian Score (LS) is fundamentally based on Laplacian Eigenmaps [107] and Locality Preserving Projection [108]. The basic idea of LS is to evaluate the features according to their locality preserving power. In Section 6.1, we re-state the algorithm to calculate the Laplacian Score as shown in [62].

The Algorithm Let $x_p = [f_{1p}, f_{2p}, f_{3p}, \ldots, f_{np}]$, be the p-th traffic sample in this chapter, where $p = 1, 2, \ldots, P$. Then, f_{ip} denotes the p-th sample of the i-th feature. Let L_i denote the Laplacian Score of the i-th feature, where $i = 1, ..., n$. The algorithm can be stated as follows.

1. Construct a nearest neighbor graph with P nodes. The p-th node is denoted by x_p. We put an edge between nodes p and q if x_p and x_q are "close", i.e. x_p is among k nearest neighbors of x_q or x_q is among k nearest neighbors of x_p. When the label information is available, one can put an edge between two nodes sharing the same label.

2. If nodes p and q are connected, put $S_{pq} = e^{-\frac{||x_p - x_q||^2}{t}}$, where t is a suitable constant. Otherwise, put $S_{pq} = 0$. The weight matrix S of the graph models the local structure of the data space.

3. For the i-th feature, we define: $\mathbf{f}_i = [f_{i1}, f_{i2}, ..., f_{iP}]^T$, $D = diag(S\mathbf{1})$, $\mathbf{1} = [1, \ldots, 1]^T$, $L = D - S$ where the matrix L is often called graph Laplacian [109]. Let

$$\check{\mathbf{f}}_i = \mathbf{f}_i - \frac{\mathbf{f}_i^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1} \tag{6.1}$$

4. Compute the Laplacian Score of the i-th feature as follows.

$$L_i = \frac{\check{\mathbf{f}}_i^T L\check{\mathbf{f}}_i}{\check{\mathbf{f}}_i^T D\check{\mathbf{f}}_i} \tag{6.2}$$

6.2 Modified Laplacian Score

To ensure the values of L_i and mutual information do not vary greatly, both values are adapted to the range [0,1]. Therefore, in this paper, a linear transformation normalisation to the value of L_i in Equation (6.2) is used as follows.

$$NL_i = \frac{L_i - L_{min}}{L_{max} - L_{min}} \tag{6.3}$$

where L_{min} and L_{max} are the minimum and maximum values of $\{L_1, L_2, ..., L_n\}$, respectively.

As discussed above, the Laplacian score does not take into consideration the redundancy among selected features. To address this issue, a scheme is proposed to eliminate redundancy among the selected features based on mutual information and appended to the Laplacian score.

Given a features set $F = \{f_1, f_2, \ldots, f_n\}$, where n is the total number of features, the task is to select the best subset of features $G = \{g_1, g_2, \ldots, g_{|G|}\}$, where $|G|$ is the number of selected features. The scheme is to normalise the value of mutual information between a candidate feature and the set of previously selected features by the entropies of the selected features as shown in Equation (6.4) in order to select the m-th feature, g_m, from $F \setminus \{g_1, g_2, ..., g_{m-1}\}$.

$$RPI(f_i; G) = \frac{1}{m-1} \sum_{j=1}^{m-1} \frac{I(f_i; g_j)}{H(g_j)}. \tag{6.4}$$

$$g_m = \underset{f_i}{\operatorname{argmax}}(NL_i - RPI(f_i; G)), \tag{6.5}$$

where NL_i represents the normalised Laplacian score of the i feature as shown in Equation (6.3).

The overall procedure of EL algorithm is as follows.

Algorithm 6 Overall procedure of EL

Input: Feature set $F = \{f_i, i = 1, ..., n\}$, R : the number of selected features, $R \leq n$.

Output: G- the selected feature subset.

1. Initialization: set $G = \phi$.

2. Calculate NL_i $(i = 1, ..., n)$ according to Equation (6.2) and Equation (6.3) for each feature in F.

3. Select the feature f_i that maximises NL_i.

Set $F \longleftarrow F \setminus \{ f_i \}$; $G \longleftarrow G \cup \{ f_i \}$.

4. while $|G| < R$ **do**

  for *each feature* $f_i \in F$ **do**

    Calculate $RPI(f_i; G)$ in Equation (6.4) for all pairs of $(f_i; G)$.

  end

  Using Equation (6.5) select g_m.

  Set $F \longleftarrow F \setminus \{ g_m \}$ *and* $G \longleftarrow G \cup \{ g_m \}$.

end

return G

We perform the unsupervised feature selection method in a different way from the

one in EL. We normalise the value of MI between candidate feature f_i in $F \setminus \{g_1, g_2, ..., g_{m-1}\}$ and the set of previously selected features, $\{g_1, g_2, ..., g_{m-1}\}$, based on the entropies of features in
$F \setminus \{g_1, g_2, ..., g_{m-1}\}$ as shown in Equation (6.6) in order to select the m-th feature, g_m, from $F \setminus \{g_1, g_2, ..., g_{m-1}\}$.

$$MRPI(f_i; G) = \frac{1}{m-1} \sum_{j=1}^{m-1} \frac{I(f_i; g_j)}{H(f_i)}, \tag{6.6}$$

where $i \in \{1, 2, \ldots, n\}$ and n is the total number of features in F.

Therefore, the main criterion of the ML is to iteratively select the feature that maximises the formula in Equation (6.7).

$$g_m = \underset{f_i}{\operatorname{argmax}}(NL_i - MRPI(f_i; G)), \tag{6.7}$$

where NL_i represents the normalised Laplacian score of the i feature as shown in Equation (6.3).

The overall procedure of ML algorithm is as follows.

Algorithm 7 Overall procedure of ML

Input: Feature set $F = \{f_i, i = 1, ..., n\}$, R : the number of selected features, $R \leq n$.

Output: G- the selected feature subset.

1. Initialization: set $G = \phi$.

2. Calculate NL_i $(i = 1, ..., n)$ according to Equation (6.2) and Equation (6.3) for each feature in F.

3. Select the feature f_i that maximises NL_i.

Set $F \longleftarrow F \setminus \{ f_i \}$; $G \longleftarrow G \cup \{ f_i \}$.

4. while $|G| < R$ **do**

  for *each feature* $f_i \in F$ **do**
    Calculate $MRPI(f_i; G)$ in Equation (6.6) for all pairs of $(f_i; G)$.
  end
  Using Equation (6.7) select g_m.
  Set $F \longleftarrow F \setminus \{ g_m \}$ *and* $G \longleftarrow G \cup \{ g_m \}$.

end

return G

6.3 Intrusion Detection Based on Unsupervised Feature Selection

The framework proposed in this chapter differs from the ones proposed in the previous chapters in the pre-selection stage, in which the proposed unsupervised feature selection is applied. The framework of the proposed detection model is shown in

Figure 6.1. It can be seen from the figure that the detection framework is comprised of four main stages:

- Data Collection. It is the first and most important stage to intrusion detection where a sequence of network packets is collected.
- Data Pre-processing. In this stage, the obtained training and test data from the data collection stage are first pre-processed to generate basic features. This phase involves three main steps. The first step is data transferring, in which every symbolic feature in a dataset is first converted into a numerical value. The second step is data normalisation, in which each feature in the data is scaled into a well-proportioned range to eliminate the bias in favour of features with greater values from the dataset. The third step is feature selection, in which the proposed feature selection algorithm is used to nominate the most

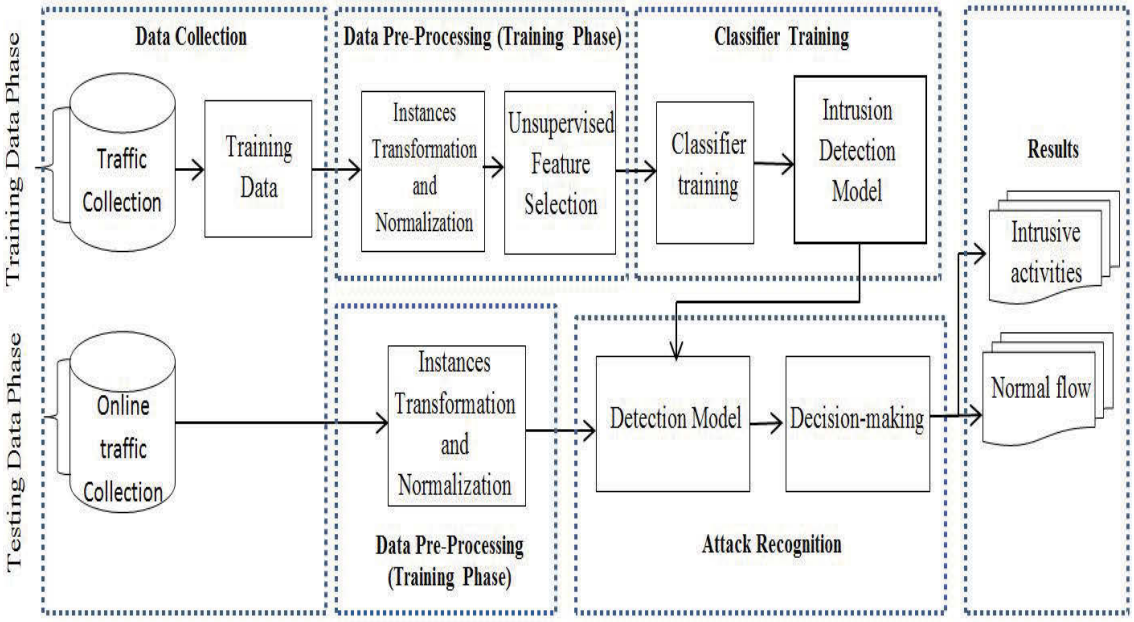

FIGURE 6.1: The framework of the proposed intrusion detection system

important features that are then used to train the classifier and build the intrusion detection model.

- Classifier Training. In this stage, the classifier is trained. Once the best subset of features is selected, this subset is then passed into the classifier training stage where a specific classification method is employed.

- Attack Recognition. In this stage, the trained model is used to detect intrusions on the test data. After completing all the iteration steps and the final classifier is trained which includes the most correlated and important features, the normal and intrusion traffics can be recognised by using the saved trained classifier. The test data is then taken through the trained model to detect attacks.

6.4 Experiments and results

To validate performance fairly, eight well-known benchmark datasets are adopted in our experiments. These eight datasets are Pen dataset, Wine dataset, Waveform dataset, Satimage dataset and Sonar dataset from the UCI Machine Learning Repository [110] and KDD Cup 99 datasets [82], NSL-KDD datasets [71] and Kyoto 2006+ datasets [72] from IDS. All of theses datasets are freely accessed. These datasets are frequently used in literature to assess and verify the performance of feature selection methods. In addition, these datasets have different types, sample sizes and various numbers of features which can provide suitable and comprehensive tests in validating feature selection algorithms. As our scheme is also studying feature behaviour and feature selection techniques, these datasets can be used to

demonstrate the validity and novelty of our scheme. Table 6.1 summarises some details and general description information about these datasets.

To evaluate the effectiveness of our proposed algorithm, binary classification and multi-classification are performed. In this paper, two classifiers are used to serve the purpose of evaluations and comparisons, and they are the nearest neighborhood classifier and Support Vector Machine (LIBSVM package [111]). These classifiers represent different learning types and are often used in literature due to their efficiency and performance. The results obtained using our proposed ML algorithm are compared against the results obtained using the EL method, Variance score method [105] and Laplacian score method [62]. The comparison results about classification accuracies on all datasets for both classifiers using these four feature selection algorithms are presented in Tables [2-5] and Figures [2-3]. In addition, the performance of our proposed feature selection algorithm is further compared with the results of the LGFS and E-LGFS methods reported in [9] on three UCI datasets, which are Pen dataset, Satimage dataset and Sonar dataset. To facilitates a fair and rational comparisons with these two methods, we select the same number of samples. Table 6.6 and Table 6.7 show the comparison results.

6.4.1 Experimental settings

During the experiments, the value of R is given by the user in advance, which represents the number of desired features. To set the best value of k, we have conducted several experiments. The optimal value of k that gives the best classification accuracy is selected. In these experiments, the value of k is selected to be 4 for both the Laplacian Score and our proposed ML algorithm. To achieve impartial results

and decrease the random selection effect, all of the experimental results presented in this paper are the averages of 10 independent runs. The best achieved results among all feature selection methods are highlighted in bold font.

To avoid the bias in favor of features with greater values in all datasets, every feature within each record is normalised by the respective maximum value and falls into the same range of [0,1].

TABLE 6.1: General information and summary of datasets used in the experiments

Dataset	# Sample	# feature	# Class	# Training	# Testing
Waveform	5000	21	3	2500	2500
Wine	178	13	3	89	89
Sonar	208	60	2	104	104
Pen	10992	16	10	2000	8992
Satimage	420	36	6	210	210
KDD Cup 99	1000	41	5	500	500
NSL-KDD	1000	41	2	500	500
Kyoto 2006+	1000	23	2	500	500

6.4.2 Benchmark Datasets

The Waveform dataset has two versions available at the UCI repository [110]. Both versions are problems with three different classes. For this study, we conduct our experiments on Waveform version 1. This version includes 5000 samples and is defined by 21 different numerical features. The sonar dataset contains information of 208 various objects with 60 features, and two different classes, rock and mine. The Wine dataset consists of 178 samples and each sample is unique with 13 features. This dataset contains three different types of wines (classes) and the goal is to

classify each one. For experimental purposes on these three datasets, we select all samples. Half of the samples are selected as training data, and the remaining are used as testing data.

The dataset Pen consists of 10,992 samples from 10 different classes and 16 features. We select all data samples in Pen dataset and all classes. We randomly select 2000 samples for training and the remaining are used as testing data. The Satimage dataset [111] consists of approximately 4435 samples with 36 features and 6 classes. We randomly select 420 samples for our experiments, half of the samples are used as training data, and the other half are used as testing data.

The KDD Cup 99 dataset is one of the most popular intrusion detection datasets and is widely applied to evaluate the performance of intrusion detection systems [22]. It consists of five different classes, which are normal and four types of attacks (i.e., DoS, Probe, U2R and R2L). The NSL-KDD is a new revised version of the KDD Cup 99 that has been proposed by Tavallaee et al. in [71]. This dataset addresses some problems included in the KDD Cup 99 dataset such as the huge number of redundant records in KDD Cup 99 data. Similar to the KDD Cup 99 dataset, each record in the NSL-KDD dataset has 41 different quantitative and qualitative features. The Kyoto 2006+ dataset was presented by Song et al. [72]. The dataset covers over three years of real traffic data collected from both honeypots and regular servers that are deployed at Kyoto University. Each connection in this dataset is unique with 23 features. For the experiments on Kyoto dataset, samples that form the data of the days 2009 August 27, 28, 29, 30 and 31 are selected and they contain the latest updated data. For experimental purposes on IDS datasets, 1000 samples from each dataset are randomly selected. Half of the samples are used as training data, and the other half is used as testing data.

6.4.3 Results on UCI datasets

The experimental results about classification accuracies on UCI datasets [110] using three different feature selection algorithms are presented in Table 6.2 and Table 6.3. The tables show the average classification accuracies using five different values of R on each dataset. As we can see in Table 6.2 and Table 6.3, based on 1NN and SVM classifiers, the results obtained using the ML and EL are better than those obtained from the Variance score and Laplacian score methods on all datasets in most of the cases. This is because both ML and EL consider the redundancies among features and they can select features with smaller redundancies.

Although, on Table 6.2, the accuracies of our method on Pen dataset when $R = 6$ and $R = 16$ are slightly lower than those using the Variance score method, they are still comparable and are better on the rest of cases. Similar case appears in Table 6.3 when $R = 16$ and SVM classifier is applied.

Considering the variances obtained using both classifiers, the proposed method has the lowest variances in most of the cases. However, in some other cases, the variances of ML are a bit higher than the best results of the other methods. For example, on Waveform dataset, using 1NN classifier (see Table 6.2), our method has achieved the lowest variances when $R = 4$, $R = 16$ and $R = 36$, while the Variance score method has achieved the lowest variances when $R = 8$ and $R = 12$. However, considering both the accuracy rates and the variances, our method has obtained better results than the Variance score method because the accuracy range of our method is higher than the Variance score method. As another example, on the Satimage dataset, although our method has bigger variances when $R = 6$, $R = 8$ and $R = 36$, it has better accuracies in all cases. Similar to the above argument on

the Waveform dataset, when taking into account of both accuracies and variances, our method always gives the best results. On Satimage datasets, using the SVM classifier (see Table 6.3), although our method has higher variances than EL when $R = 2$, $R = 6$ and $R = 8$, it has better accuracies in all cases and has better results when taking into account both accuracies and variances.

Note that when R is increased to n (i.e., when all featues are used) from the last selected numbers, the accuracy rates obtained using either ML or EL do not show big improvements when testing on most of the datasets, and in some of the datasets the accuracy rates are fluctuated. This concludes that many features are not important to the classification processes and they may only be redundant features. For example, in the case of Sonar dataset when using ML and using the 1NN classifier (see Table 6.2), there is only a slight increase (from 82.42% to 82.88%) on the classification accuracies when R is increased to 60 (i.e., the maximum value) from the last selected value, 40. On the Waveform dataset, when using the 1NN classifier (see Table 6.2), our method has achieved an accuracy of 78.64% when $R = 16$ (the last selected) and the accuracy decreases to 77.61% when $R = 21$ (the maximum feature number).

Figure 6.2 plots the classification accuracies of 1NN and SVM classifiers achieved using ML, Variance score and Laplacian score methods with R increasing from 1 to n. The x axis represents the number of selected features and the y axis represents the classification accuracy. The results shown in the figure are based on three UCI datasets: Pen, Wine and Waveform datasets. The figure shows that, in general, the classification accuracy improves when the number of selected features increases. It can be seen from the figure that the curve of ML is above the curves of other

TABLE 6.2: A comparison of classification accuracies using three feature selection algorithms on UCI datasets based on 1NN

#R	Variance score	Laplacian Score	EL	Proposed ML
Wine ($n = 13$)				
3	71.42 ± 9.45	83.93 ± 7.36	84.49 ± 6.82	**88.60 ± 3.12**
6	87.98 ± 7.14	94.49 ± 4.30	95.39 ± 3.77	**95.77 ± 2.53**
9	94.04 ± 5.93	95.28 ± 4.29	95.16 ± 3.18	**95.84 ± 2.54**
12	95.58 ± 3.96	95.69 ± 2.75	95.62 ± 2.05	**95.81 ± 1.78**
13	**95.51** ± 2.82	**95.51** ± 1.76	**95.51** ± 1.88	**95.51 ± 1.64**
Pen ($n = 16$)				
3	60.39 ± 1.30	60.85 ± 0.55	65.99 ± **0.24**	**66.37** ± 1.29
6	**88.44 ± 0.46**	87.01 ± 1.03	87.28 ± 1.83	88.27 ± 0.91
9	95.22 ± 0.07	94.29 ± 1.19	94.37 ± 0.81	**95.53 ± 0.06**
12	96.79 ± 0.08	96.86 ± 0.041	96.93 ± **0.05**	**97.20** ± 0.06
16	**98.53** ± 0.03	98.50 ± 0.03	98.52 ± **0.02**	98.52 ± **0.02**
Waveform ($n = 21$)				
4	62.17 ± 3.83	61.43 ± 3.31	62.67 ± 3.95	**64.46 ± 2.81**
8	71.12 ± **1.33**	70.49 ± 4.47	70.83 ± 2.55	**72.84** ± 2.55
12	77.86 ± **0.41**	77.29 ± 0.66	**78.21** ± 0.42	**78.21** ± 0.42
16	77.77 ± **0.53**	77.94 ± 0.63	78.53 ± 0.67	**78.64 ± 0.53**
21	77.26 ± 0.29	77.29 ± 0.28	77.33 ± 0.25	**77.61 ± 0.15**
Satimage ($n = 36$)				
2	85.14 ± 4.63	87.62 ± 4.18	88.38 ± 2.53	**88.76 ± 2.03**
4	90.43 ± 3.60	89.86 ± 5.39	90.81 ± 5.44	**90.86 ± 3.11**
6	91.19 ± 2.28	91.05 ± 2.86	91.43 ± **2.21**	**92.29** ± 2.42
8	91.81 ± 2.70	91.52 ± 2.86	91.77 ± **1.62**	**92.24** ± 1.67
36	94.09 ± 1.56	94.33 ± **0.83**	**94.38** ± 0.99	**94.38** ± 0.99
Sonar ($n = 60$)				
10	69.13 ± 9.99	72.31 ± 9.18	73.17 ± 8.63	**73.27 ± 7.42**
20	74.04 ± 7.82	78.94 ± 6.88	79.04 ± 6.86	**79.81 ± 6.65**
30	79.13 ± **5.32**	79.42 ± 7.64	79.53 ± 6.18	**79.81** ± 6.01
40	80.19 ± 5.56	81.64 ± 5.20	81.64 ± 5.53	**82.42 ± 5.02**
60	83.49 ± 3.16	83.27 ± 3.06	**83.88 ± 3.04**	**83.88 ± 3.04**

methods for almost all R values, to the performance of ML method is better than those of Laplacian score and Variance score methods in all of the cases.

TABLE 6.3: A comparison of classification accuracies using three feature selection algorithms on seven datasets based on SVM

#R	Variance score	Laplacian Score	EL	Proposed ML
Wine ($n = 13$)				
3	72.13 ± 9.26	85.40 ± 4.22	87.07 ± 4.03	**88.76 ± 2.21**
6	90.11 ± 4.86	93.93 ± 3.98	94.42 ± 2.59	**94.88 ± 2.11**
9	94.94 ± 3.49	95.96 ± 2.58	96.74 ± 2.33	**97.01 ± 1.14**
12	96.67 ± 1.69	96.85 ± 1.46	97.30 ± 2.58	**97.80 ± 0.92**
13	97.60 ± 1.87	97.64 ± 1.80	**97.64 ± 1.53**	**97.64 ± 1.53**
Pen ($n = 16$)				
3	66.09 ± 1.14	67.29 ± 1.34	70.74 ± **0.07**	**71.35** ± 1.49
6	87.53 ± 2.59	88.53 ± 3.04	89.11 ± 2.74	**90.37 ± 2.42**
9	95.75 ± 0.39	95.48 ± 0.49	95.78 ± 0.43	**96.05 ± 0.33**
12	97.92 ± 0.05	98.02 ± 0.05	98.04 ± 0.04	**98.13 ± 0.02**
16	**99.03 ± 0.02**	99.02 ± **0.02**	99.02 ± **0.02**	99.02 ± **0.02**
Waveform ($n = 21$)				
4	69.29 ± 2.02	69.64 ± 1.17	69.81 ± 2.22	**71.40 ± 1.05**
8	76.63 ± 1.31	76.92 ± 2.54	77.02 ± 1.08	**77.66 ± 1.04**
12	82.85 ± 0.55	82.84 ± **0.60**	83.04 ± **0.60**	**83.15** ± 0.61
16	82.84 ± **0.11**	83.24 ± 0.62	83.52 ± 0.27	**83.65** ± 0.24
21	83.21 ± 0.38	83.35 ± 0.30	83.37 ± 0.23	**83.43 ± 0.23**
Satimage ($n = 36$)				
2	83.95 ± 5.85	84.33 ± 4.39	86.76 ± **3.24**	**87.19** ± 3.65
4	88.71 ± 4.96	89.67 ± 3.91	90.24 ± 4.88	**91.05 ± 3.89**
6	90.67 ± 4.25	90.29 ± 4.29	91.67 ± **1.32**	**92.19** ± 2.83
8	92.62 ± 2.43	92.90 ± 2.69	93.28 ± **2.04**	**93.48** ± 2.52
36	94.38 ± 1.60	94.28 ± 1.01	**94.43 ± 0.71**	**94.43 ± 0.71**
Sonar ($n = 60$)				
10	71.63 ± 8.84	71.44 ± 8.25	**75.87** ± 7.65	**75.87 ± 7.62**
20	77.12 ± 7.24	77.88 ± 6.79	78.75 ± 7.34	**78.85 ± 5.32**
30	76.35 ± 4.75	78.08 ± 4.37	79.03 ± 4.06	**79.13 ± 3.34**
40	74.33 ± 3.87	80.10 ± 3.24	80.29 ± 3.05	**80.29 ± 2.95**
60	**83.46 ± 2.05**	**83.46 ± 2.05**	**83.46 ± 2.05**	**83.46 ± 2.05**

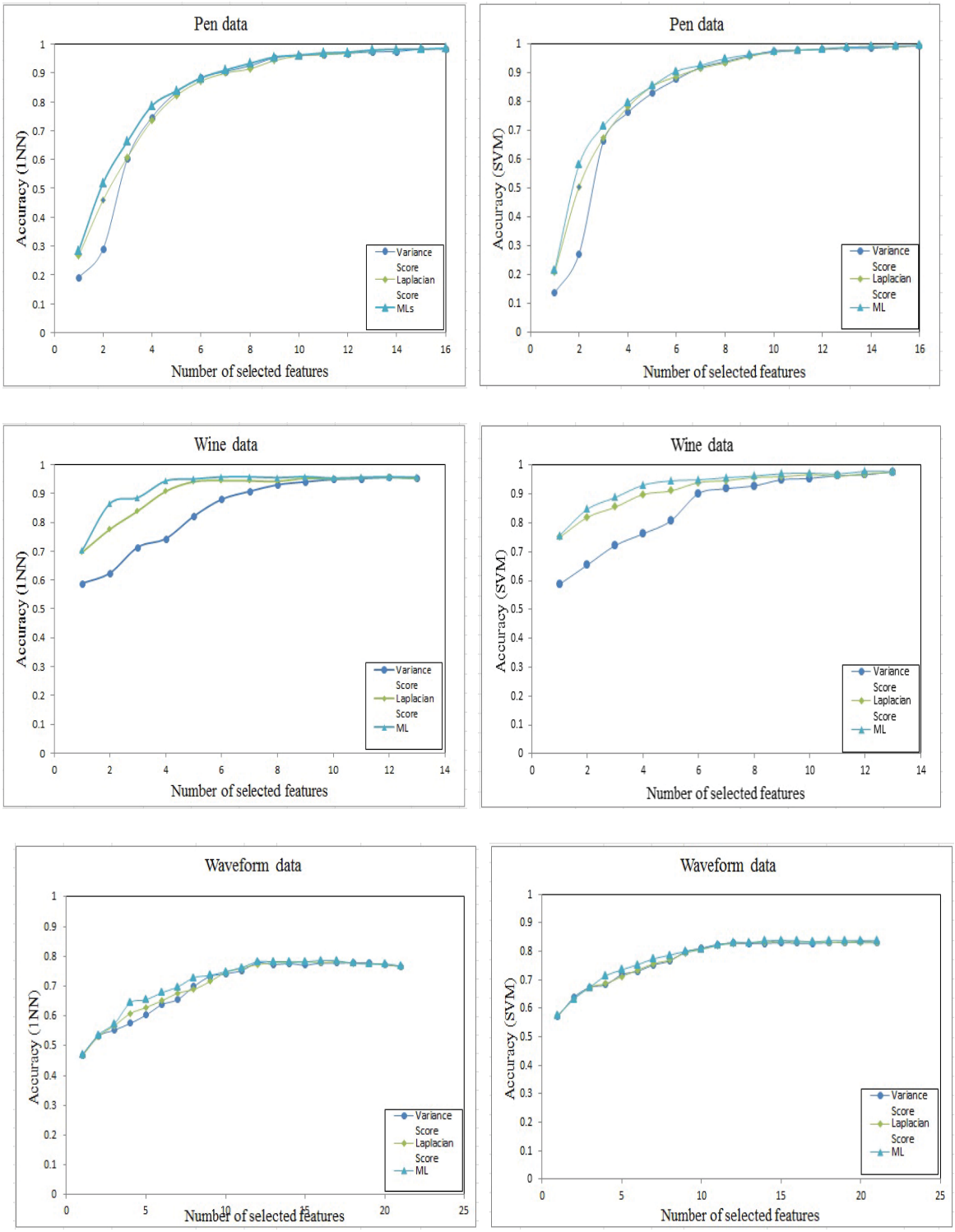

FIGURE 6.2: Effect of number of selected features on UCI datasets

6.4.4 Results on IDS datasets

In order to further investigate the performance of the proposed feature selection algorithm, three intrusion detection datasets are used. The aim is to further examine the advantages of removing redundancies among the selected features. The experimental results on the IDS datasets, which are again based on 1NN and SVM classifiers, using our algorithm (i.e., ML), EL, Laplacian score and Variance score are presented in Table 6.4, Table 6.5 and Figure 6.3.

Through Table 6.4 and Table 6.5, the accuracies with bold font represent the highest accuracies of the four comparing algorithms. The results in both tables are the average classification accuracies of five different values of R on each dataset. It can be seen clearly that the classification accuracies obtained by both classifiers using our ML method are the best compared to those obtained using other methods in all datasets. In addition, the proposed ML outperforms EL using both classifiers on all datasets. Considering the variances obtained using both classifiers on all IDS datasets, the proposed method has achieved the lowest variances in most of the cases. In some cases, such as on the KDD dataset when $R = 8$, our method has a bit higher variance than EL when using 1NN classifier (see 6.4). However, considering both the accuracies and variances, our method still performs better than EL because the accuracy range using ML is still higher than that of EL. Similarly, when tested on the Kyoto dataset with $R = 12$ using 1NN and on the NSL dataset with when $R = 16$ using SVM, our method has higher variances but overall has better accuracies than EL.

Figure 6.3 shows the classification accuracies using the 1NN and SVM classifiers achieved by the three algorithms with R increasing from 1 to n. As it can be seen

TABLE 6.4: A comparison of classification accuracies using three feature selection algorithms on IDS datasets based on 1NN

#R	Variance score	Laplacian Score	EL	Proposed ML
KDD ($n = 41$)				
4	75.18 ± 9.17	79.54 ± 9.39	87.12 ± 8.75	**89.22 ± 7.19**
8	89.48 ± 4.72	87.44 ± 4.95	89.08 ± **4.54**	**89.92** ± 4.68
12	93.24 ± 3.71	92.4 ± 4.50	93.58 ± 3.32	**94.32 ± 2.61**
16	97.27 ± 1.59	97.18 ± 2.01	97.18 ± 1.33	**97.38 ± 0.47**
41	**99.58 ± 0.18**	**99.58 ± 0.18**	**99.58 ± 0.18**	**99.58 ± 0.18**
NSL ($n = 41$)				
4	62.76 ± 8.72	60 ± 9.35	60.96 ± 9.03	**68.76 ± 8.33**
8	80.04 ± 6.24	75.58 ± 6.47	79.58 ± 6.76	**84.86 ± 5.92**
12	89.22 ± 3.50	89.20 ± 3.28	90.54 ± 3.30	**91.96 ± 3.09**
16	95.16 ± 1.49	95.10 ± 1.02	95.24 ± 2.01	**95.42 ± 0.93**
41	**97.28 ± 0.27**	97.27 ± **0.27**	**97.28 ± 0.27**	**97.28 ± 0.27**
Kyoto ($n = 23$)				
3	79.84 ± 11.56	82.6 ± 9.95	86.14 ± 9.74	**88.36 ± 9.46**
6	92.26 ± 2.98	94.20 ± 2.01	94.66 ± 6.37	**95.42 ± 1.55**
9	96.54 ± 0.98	96.66 ± 0.61	96.78 ±0.48	**96.94 ± 0.25**
12	96.64 ± 0.60	96.78 ± 0.69	97.06 ± **0.56**	**97.56** ± 0.64
23	97.38 ± 0.47	97.22 ± 0.41	**97.44 ± 0.41**	**97.44 ± 0.41**

from the figure, in general, the accuracies improve when the number of selected features increases. In addition, the results show that the accuracies based on both classifiers using our algorithm is better than those using other methods.

6.4.5 Comparison with LGFS and E-LGFS

In the following experiments, we compare the results obtained using our algorithm with the results achieved in [9] for LGFS and E-LGFS methods on three UCI datasets: Pen, Satimage and Sonar dataset. Similar to LGFS and E-LGFS, for experiments on the Pen dataset, we select 300 samples from classes 3, 8 and 9.

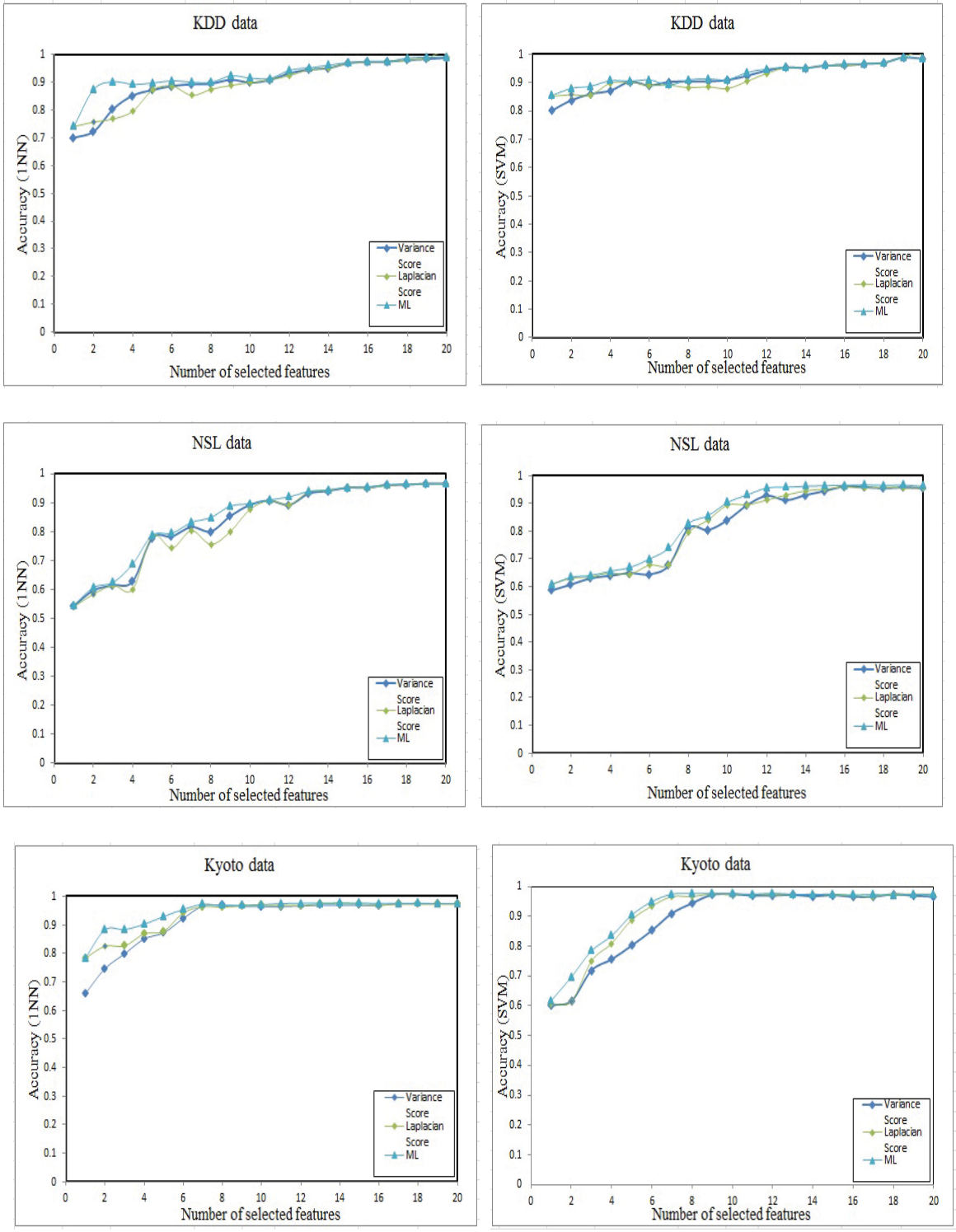

FIGURE 6.3: Effect of number of selected features on IDS datasets with the two classifiers

TABLE 6.5: A comparison of classification accuracies using three feature selection algorithms on seven datasets based on SVM

#R	Variance score	Laplacian Score	EL	Proposed ML
KDD ($n = 41$)				
4	87.15 ± 9.19	89.8 ± 7.56	89.8 ± 8.13	**90.7 ± 7.04**
8	90.39 ± 6.83	88.22 ± 6.96	90.4 ± 6.74	**90.94 ± 6.42**
12	94.26 ± 2.06	93.26 ± 3.98	94.10 ± 3.43	**94.74 ± 1.60**
16	96.38 ± 1.22	96.06 ± 1.51	96.10 ± 1.25	**96.56 ± 1.17**
41	**99.08 ± 0.22**	**99.08 ± 0.22**	**99.08 ± 0.22**	**99.08 ± 0.22**
NSL ($n = 41$)				
4	63.91 ± 8.27	64.87 ± 8.12	65.18 ± 8.19	**65.56 ± 8.07**
8	81.09 ± 7.16	79.4 ± 6.33	82.70 ± 6.38	**82.76 ± 6.25**
12	92.71 ± 4.72	91.14 ± 5.71	93.32 ± 5.29	**95.50 ± 3.24**
16	96.10 ± 0.87	95.86 ± 0.91	95.88 ± **0.74**	**96.44** ± 0.77
41	96.51 ± 0.55	96.52 ± 0.53	**96.52 ± 0.51**	**96.52 ± 0.51**
Kyoto ($n = 23$)				
3	71.82 ± 9.38	74.94 ± 7.63	76.42± 7.42	**78.38 ± 7.26**
6	85.23 ± 6.98	93.40 ± 2.61	93.46 ± 2.29	**94.90 ± 1.25**
9	97.22 ± 1.81	97.54 ± **0.40**	97.44 ± 0.72	**97.66** ± 0.84
12	96.98 ± 0.53	97.18 ± 0.89	97.36 ± 0.58	**97.72 ± 0.47**
23	97.60 ± 0.58	97.40 ± 0.62	97.68 ± **0.54**	**97.74 ± 0.54**

From each class, we randomly select 100 samples. Again Similar to LGFS and E-LGFS, we select all available samples from the Sonar datasets and randomly select 420 samples from the Satimage dataset.

Table 6.6 and Table 6.7 summarize the average classification accuracies obtained using 1NN and SVM classifiers of the three feature selection algorithms. The results in both tables are based on 4 different values of R on each dataset. The best results achieved among these feature selection methods are emphasized in bold font. The results presented in both tables are the averages of 50 independent runs.

From the results in Table 6.6 and Table 6.7, one can observe that both E-LGFS and the proposed ML enjoy the best classification accuracies on all datasets in

TABLE 6.6: A comparison of classification accuracies using three feature selection algorithms on four UCI datasets based on 1NN

#R	LGFS	E-LGFS	Proposed ML
Pen ($n = 16$)			
2	81.91 ± 6.88	**84.56 ± 4.25**	82.33 ± 5.11
5	95.88 ± **1.52**	96.32 ± 1.59	**96.43** ± 1.53
8	96.33 ± 1.41	97.11 ± 1.49	**98.45 ± 0.97**
11	98.08 ± 1.26	98.11 ± 1.35	**99.19 ± 0.55**
Satimage ($n = 36$)			
6	85.34 ± 2.23	85.88 ± **1.99**	**91.46** ± 2.23
12	88.85 ± 2.35	89.36 ± **2.28**	**92.16** ± 2.84
18	90.20 ± 1.96	90.70 ± 1.97	**92.83 ± 1.76**
24	90.80 ± 1.78	91.51 ± 1.77	**93.62 ± 1.77**
Sonar ($n = 60$)			
10	74.46 ± **4.60**	**76.23** ± 4.67	75.96 ± 4.64
20	80.59 ± 3.60	81.22 ± 3.46	**81.73 ± 3.25**
30	81.16 ± 3.78	**81.64 ± 3.56**	81.58 ± 3.82
40	82.59 ± **3.66**	82.60 ± **3.66**	**82.62** ± 3.68

most of the cases. This can be regarded to the fact that these two methods take into consideration the redundancies among features during the selection processes. Therefore, features with high redundancies will be neglected. In contrast, features that can provide useful information to the classification have high probabilities to be selected by both algorithms.

Compared to E-LGFS, our method performs better in most of the cases, and in the other cases the differences are very small. For example, on the Sonar dataset, our method has obtained comparable results with those of E-LGFS. However, the results obtained by ML on the Pen and Satimage datasets are much better than those obtained using E-LGFS. In addition, although our method has achieved lower accuracy on the Pen dataset when $R = 2$ using 1NN (see Table 6.6) than E-LGFS,

TABLE 6.7: A comparison of classification accuracies using three feature selection algorithms on four UCI datasets based on SVM

#R	LGFS	E-LGFS	Proposed ML
Pen ($n = 16$)			
2	85.96 ± 6.73	**88.04 ± 3.67**	86.93 ± 5.73
5	94.63 ± **1.40**	95.93 ± 1.81	**96.17** ± 1.68
8	96.55 ± 1.78	97.07 ± 1.71	**98.76 ± 0.65**
11	97.72 ± 1.04	97.80 ± 1.02	**99.44 ± 0.22**
Satimage ($n = 36$)			
6	83.54 ± 2.84	84.35 ± **2.35**	**92.18** ± 3.16
12	87.28 ± 1.86	87.55 ± **1.68**	**92.78** ± 3.78
18	88.26 ± **1.40**	88.81 ± 1.69	**93.23** ± 1.68
24	88.69 ± **1.56**	89.18 ± 1.61	**93.49** ± 1.79
Sonar ($n = 60$)			
10	73.24 ± 6.02	74.17 ± 5.75	**74.46 ± 5.61**
20	76.40 ± 4.99	77.33 ± 4.28	**79.96 ± 4.09**
30	79.20 ± 3.71	79.87 ± 3.56	**80.76 ± 3.49**
40	81.30 ± 3.79	81.39 ± 3.69	**81.43 ± 3.38**

it outperforms E-LGFS for bigger R values. Similarly, although the variances of our method are higher on the Satimage dataset using SVM (see Table 6.7), overall our method has achieved better accuracies in all cases.

To sum up, the results obtained form the three datasets indicate that eliminating redundancies among features improves the classification performance of classifiers. In addition, as it has been claimed in [62, 106], extracting the local structure information of features in some applications may be enough to select the best subset of features and achieve promising classification performance.

6.5 Summary

This Chapter has proposed an unsupervised feature selection algorithm, which is an enhancement over the Laplacian score method. The algorithm is named a Modified Laplacian score, ML in short. More specifically, two main phases are involved in ML during the selection processes. In the first phase, a k-nearest neighbor graph is used to capture the locality preserving power of each feature. In the second phase, a Redundancy Penalization (RP) function is used to eliminate redundancies among the selected features. RP is based on the principle of mutual information. This method is a modified version of our work EL proposed in a paper accepted by TrustCom-2015 conference. ML differs from that of EL in the redundancy measure technique. ML proposes a measure of redundancy between the candidate feature f_i and the subset of previously selected features g_j (as shown in Equations (6.6) and Equation (6.7). The measure is to normalise the value of mutual information between f_i and g_j by the entropy of the candidate feature f_i instead of the entropy of previously selected features g_j.

To investigate the effectiveness of the proposed method, several experiments have been conducted on five UCI datasets and three IDS datasets. The performance of ML is compared against the results obtained using EL, Laplacian score and Variance score methods. Experimental results have shown that our method has achieved encouraging results and outperformed the Laplacian score and Variance score algorithms in terms of classification accuracies in almost all cases. Compared to the EL method, using 1NN and SVM classifiers, the proposed ML enjoys better classification accuracies on the utilized datasets in almost all of the cases. In addition, compared with the LGFS and E-LGFS methods, using both classifiers, ML

has achieved the best accuracies in most of cases when tested on the Pen, Waveform and Sonar datasets.

Chapter 7

Conclusion and Future Work

7.1 Summary of Contributions

Intrusion Detection is a well-established and active field of research in computer security and networks. Intrusion detection systems play a critical role in securing the communications infrastructure of most organisations. This thesis has proposed novel frameworks addressing three significant issues that severely affect the performance and utility of the present intrusion detection systems. The three issues are:

- Large number of false alarms,
- High volume network traffic, and
- The classification problem when the class labels are not available.

To address these three issues, we have conducted in-depth research on IDS and developed efficient detection models to detect a variety of attacks with very few false alarms and low computational costs. An introduction to the works presented in this thesis have been given in Chapter 2. The chapter has briefly presented the concept of anomaly-based intrusion detection with existing detection techniques. Then, it has presented a review of some of the advantages utilising dependency measures in enhancing the detection performance and reducing the false alarm rates of anomaly detection systems. An overview of feature selection and the most related feature selection methods relevant to this thesis have been presented. Two main categories of feature selection techniques have been reviewed in this chapter: the supervised feature selection and the unsupervised feature selection. The chapter has also analysed and outlined the limitations of the existing feature selection algorithms. A summary of the contributions conducted in this thesis is given in the following.

- Chapter 3 has introduced a Nonlinear Correlation Coefficient (NCC) based on a similarity measure for extracting the relationship between network traffic records. NCC is designed based on the definition of mutual information, which is capable of extracting both linear and nonlinear correlation. Then, the extracted information is used to build an IDS to detect abnormal behaviours. The detection framework proposed in this chapter is comprised of four main stages: the data collection stage, where a sequence of network packets is collected; the data preprocessing stage, where training and test data are preprocessed and important features that can distinguish one class from the others are selected; the classifier training stage, where the model for classification is trained; and the attack recognition stage, where the trained classifier

is used to detect intrusions on the test data. The proposed NCC-based intrusion detection system has achieved a lower false positive rate and a higher detection rate compared to some of the state-of-the-art detection systems.

- Chapter 4 has presented a filter-based supervised feature selection algorithm, called Flexible Mutual Information Feature Selection (FMIFS), to cope with the issue of large-scale data. FMIFS uses a mutual information method as an evaluation criterion to measure the relevance between the input features and the output classes. FMIFS introduces a new criterion to eliminate the redundancy among selected features with respect to the already selected subset of features. The proposed FMIFS is an enhanced version of Battiti's MIFS and Amiri's MMIFS. It eliminates the redundancy parameter β required by MIFS and MMIFS. This is very useful in practice since the selection of an appropriate value for this parameter is still an open question. FMIFS is then combined with the LS-SVM method to build an intrusion detection system. LS-SVM is a least square version of SVM that works with equality constraints instead of inequality constraints in the formulation to solve a set of linear equations for classification problems rather than a quadratic programming problem. The combined detection model has shown an improvement in building and testing time in comparison with those systems that need to examine all features. In addition, the performance of the proposed system exhibited better classification rates and promising results in terms of classification accuracy, detection rate, false positive rate and F-measure than the other existing related approaches.

- Chapter 5 has proposed a hybrid feature selection algorithm to enhance the classification accuracy of the FMIFS proposed in Chapter 4. Two main stages

are involved in this method: the filter feature ranking and the wrapper-based Improved Forward Floating Selection (IFFS) using LS-SVM and classification accuracy. The filter feature ranking stage is used as an upper phase to reduce the computational cost of the lower phase, where the wrapper method is applied, by eliminating noisy features from the original feature set. The lower phase chooses the optimal subset of features that produce the best classification performance by calculating the accuracy of the current selected subset and comparing it with the previously selected one. The wrapper method consists of two steps: backtracking step, which is used to avoid the nesting problem, and replacing the weak features step, which is used to check if the replacements can provide a better subset. The proposed feature selection algorithm has been assessed through building an IDS. The developed detection model has exhibited a promising results in terms of classification accuracy, low computational cost and F-measure.

- Chapter 6 has proposed unsupervised feature selection algorithms, which are an enhancement of the Laplacian score method, named an Extended Laplacian score EL and a Modified Laplacian score ML. The proposed EL and ML consist of two main stages. The k-nearest neighbor graph is applied in the first phase to extract the locality preserving power of each feature. In the second phase, a new redundancy penalization method has been used to remove redundancies among the selected features. The redundancy penalization is based on the principle of the mutual information and the entropy. The proposed algorithms EL and ML have shown that extracting the locality structure information of samples with removing redundancy among selected features achieves promising results. The final output of these algorithms is

then used to build the detection model. The experimental results on three well-known IDS datasets have shown that the IDS with EL or ML proposed in this study achieves better classification accuracy than the one with the Laplacian score Variance score methods.

7.2 Future work

Based on the results obtained in this thesis, some future extensions to the proposed research work are summarised in the following points.

- The feature selection methods presented in this thesis evaluate features individually using a specific evaluation criterion. However, evaluating a combination of features each time will be very useful in reducing the computational costs of feature selection methods. In particular, features can be evaluated jointly instead of individually. In future, this idea will be further investigated.
- The proposed FMIFS and ML feature selection algorithms are greedy selection methods, in which the number of desired features needs to be predefined. In the experiments, to set this value, the best feature set which yields the best classification accuracy is always selected. However, this requires the examination of the total range of feature set size to identify the best one. Therefore, in order to make these methods applicable to real situations, the number should be be automatically determined.
- The proposed feature selection algorithm in Chapter 6 has shown good efficiency. However, it could be further enhanced. For example, adoptive learning algorithms can be used to select an appropriate value for the parameter k.

This will be very useful since the proposed method is sensitive to the selection of this parameter. This will be considered when working on enhancements to the method.

- One of the problems that lead to degrade the performance of the detection model is the unbalanced sample distribution on the available IDS datasets. This issue creates another challenging task for IDSs. Therefore, it needs to be carefully studied in future research.

Appendix A

Least Squares Support Vector Machine

Given a training dataset with M data points $\{(x_r, y_r)\}_{r=1}^{M}$, where $x_r \in R^n$ is a n-dimensional feature vector (i.e., the $r-th$ data point) and $y_r \in R$ indicates the class to which the point belongs, the LS-SVM can be defined in Equation (A.1).

$$y(x) = w^T \vartheta(x_r) + b, \tag{A.1}$$

where w and b are two parameters, and $\vartheta(.)$ is a mapping function transforming a data point into a higher dimensional data space. The classification problem in LS-SVM is defined as the optimisation problem shown in Equation (A.2).

$$\min_{w,b,e} j(w, b, e) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{r=1}^{M} e_r^2, \tag{A.2}$$

subject to $\quad y_r = \omega^T \varphi(x_r) + b + e_r, \quad r = 1, ..., M.$

To find the optimal control law, one defines the Lagrangian as given in Equation (A.3).

$$\mathcal{L}(w, b, e; \alpha) = j(w, b, e) - \sum_{r=1}^{M} \alpha_r \{y_r [w^T \varphi(x_r) + b] - 1 + e_r\}, \tag{A.3}$$

with the Lagrange multipliers $\alpha_r \in R$. The conditions for optimality are defined in Equation (A.4).

$$\begin{aligned}
&\frac{\delta \mathcal{L}}{\delta w} = 0 \longrightarrow \omega = \sum_{r=1}^{M} \alpha_r y_r \varphi(x_r), \\
&\frac{\delta \mathcal{L}}{\delta b} = 0 \longrightarrow \sum_{r=1}^{M} \alpha_r y_r = 0, \\
&\frac{\delta \mathcal{L}}{\delta e_r} = 0 \longrightarrow \alpha_r = \gamma e_r, r = 1, ..., M, \\
&\frac{\delta \mathcal{L}}{\delta \alpha_r} = 0 \longrightarrow y_r [w^T \varphi(x_r) + b] - 1 + e_r = 0, r = 1, ..., M,
\end{aligned} \tag{A.4}$$

and they it can be rewritten as the solution to the set of linear equations shown in Equation (A.5).

$$\left[\begin{array}{ccc|c} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & \gamma I & -I \\ \hline Z & Y & I & 0 \end{array}\right] \left[\begin{array}{c} w \\ b \\ e \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \vec{1} \end{array}\right], \tag{A.5}$$

where $z = [\vartheta(x_1)^T y_1, ..., \vartheta(x_M)^T y_M], \quad Y = [y_1, ..., y_M]$, $\vec{1} = [1, ..., 1]$, $e = [e_1, ..., e_M]$ and $\alpha = [\alpha_1, ..., \alpha_M]$. After eliminating w and e, the solution can be simplified as

Equation (A.6).

$$\left[\begin{array}{c|c} 0 & -Y^T \\ \hline Y & ZZ^T + \gamma^{-1} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \vec{1} \end{array}\right]. \tag{A.6}$$

Mercer's condition can be used again to the matrix $\Omega = ZZ^T$, where

$$\begin{aligned} \Omega_{rl} &= y_r y_l \vartheta(x_r)^T \vartheta(x_l) \\ &= y_r y_1 \psi(x_r, x_l). \end{aligned} \tag{A.7}$$

Therefore, the classifier Equation (A.1) is found by solving the linear set of Equation (A.6)- Equation (A.7) instead of quadratic programming. Interested readers please refer to [79] for more details about the estimation of the various parameters.

Appendix B

Estimating Mutual Information

Assume that a set of M input-output pairs $z_i = (u_i, v_i)$, where $i = 1, ..., M$, are considered to be realisations of an i.i.d (independent and identically distributed) random variable of a random variable $Z = (U, V)$ with density $P_{U,V}(u, v)$. Both U and V have values in the data space R or in R^P. The Euclidean norm is then used in those spaces.

Input-output pairs are compared through the maximum norm

$$\| z - z' \|_\infty = max(\| u - u' \|, \| v - v' \|). \tag{B.1}$$

Considering k as a fixed positive integer, $z_{k(i)} = (u_{k(i)}, v_{k(i)})$ is the k-th nearest neighbor of z_i, which has the maximum norm. $u_{k(i)}$ and $v_{k(i)}$ denote the input and output of $z_{k(i)}$, respectively. Let us denote by $\epsilon_i/2$ the distance from z_i to its k-th neighbor and by $\epsilon_i^u/2$ and $\epsilon_i^v/2$ the distances between the same points projected into the U and V subspaces, where,

$$\epsilon_i/2 = \| z_i - z_{ki} \|_\infty, \tag{B.2}$$

$$\epsilon_i^u/2 = \| u_i - u_{ki} \|, \quad \epsilon_i^v/2 = \| v_i - v_{ki} \| . \tag{B.3}$$

Obviously, $\epsilon_i = max(\epsilon_i^u, \epsilon_i^v)$. Then, we count the number of samples m_i^u of points u_j whose distances from u_i is strictly less than $\epsilon_i/2$, and similarly for v. Mutual information can then be estimated by:

$$MI(U;V) = \psi(k) - \frac{1}{k} - \frac{1}{M}\sum_{i=1}^{M}[\psi(m_i^u) - \psi(m_i^v)] + \psi(M) \tag{B.4}$$

where ψ is the digamma function and given by:

$$\psi(m) = \Gamma(m)^{-1}\frac{d\Gamma(m)}{dm}, \tag{B.5}$$

where

$$\Gamma(m) = \int_0^\infty u^{m-1}e^{-u}du \tag{B.6}$$

The accuracy of this estimator is closely related to the value chosen for k. A small value of k leads to a large variance and a small bias of the estimator, vice versa.

Bibliography

[1] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, N. Yazdani, Mutual information-based feature selection for intrusion detection systems, Journal of Network and Computer Applications 34 (4) (2011) 1184–1199.

[2] X. Guan, W. Wang, X. Zhang, Fast intrusion detection based on a non-negative matrix factorization model, Journal of Network and Computer Applications 32 (1) (2009) 31–44.

[3] G. Vigna, R. A. Kemmerer, Netstat: A network-based intrusion detection approach, in: Computer Security Applications Conference, IEEE, 1998, pp. 25–34.

[4] A. Hassanzadeh, B. Sadeghian, Intrusion detection with data correlation relation graph, in: Availability, Reliability and Security, IEEE, 2008, pp. 982–989.

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

[6] S. Jin, D. S. Yeung, X. Wang, Network intrusion detection in covariance feature space, Pattern Recognition 40 (8) (2007) 2185–2197.

[7] C.-F. Tsai, C.-Y. Lin, A triangle area based nearest neighbors approach to intrusion detection, Pattern Recognition 43 (1) (2010) 222–229.

[8] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, F. Tang, Discriminating ddos attacks from flash crowds using flow correlation coefficient, IEEE Transactions on Parallel and Distributed Systems 23 (6) (2012) 1073–1080.

[9] Y. Ren, G. Zhang, G. Yu, X. Li, Local and global structure preserving based feature selection, Neurocomputing 89 (2012) 147–157.

[10] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, Bayesian event classification for intrusion detection, in: Computer Security Applications Conference, IEEE, 2003, pp. 14–23.

[11] S.-W. Lin, K.-C. Ying, C.-Y. Lee, Z.-J. Lee, An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection, Applied Soft Computing 12 (10) (2012) 3285–3290.

[12] N. Ye, Y. Zhang, C. M. Borror, Robustness of the markov-chain model for cyber-attack detection, IEEE Transactions on Reliability 53 (1) (2004) 116–123.

[13] A. M. Ambusaidi, L. F. Lu, X. He, Z. Tan, A. Jamdagni, P. Nanda, A nonlinear correlation measure for intrusion detection, in: International Conference on Frontier of Computer Science and Technology, IEEE, 2012, pp. 1–7.

[14] A. M. Ambusaidi, Z. Tan, X. He, P. Nanda, L. F. Lu, A. Jamdagni, Intrusion detection method based on nonlinear correlation measure, International Journal of Internet Protocol Technology 8 (2) (2014) 77–86.

[15] A. M. Ambusaidi, X. He, Z. Tan, P. Nanda, Building an intrusion detection system using a filter-based feature selection algorithm, IEEE Transactions on Computers, Under review.

[16] A. M. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, T. U. Nagar, A novel feature selection approach for intrusion detection data classification, in: International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2014, pp. 82–89.

[17] A. M. Ambusaidi, X. He, P. Nanda, Unsupervised feature selection method for intrusion detection system, in: International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2015.

[18] A. M. Ambusaidi, X. He, P. Nanda, J. Chen, Unsupervised feature selection method for intrusion detection system, Journal of Network and Computer Applications, to be submitted, 2015.

[19] F. Edgeworth, Xli. on discordant observations, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 23 (143) (1887) 364–375.

[20] V. Kumar, Parallel and distributed computing for cybersecurity, IEEE Distributed Systems Online 6 (10) (2005) 1–9.

[21] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, The International Journal on Very Large Data Bases 16 (4) (2007) 507–521.

[22] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, Intrusion detection by machine learning: A review, Expert Systems with Applications 36 (10) (2009) 11994–12000.

[23] E. Eskin, Anomaly detection over noisy data using learned probability distributions, in: International Conference on Machine Learning, 2000, pp. 255–262.

[24] W. Hu, Y. Liao, V. R. Vemuri, Robust support vector machines for anomaly detection in computer security, in: ICMLA, 2003, pp. 168–174.

[25] S. Mukkamala, A. H. Sung, A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, Journal of network and computer applications 28 (2) (2005) 167–182.

[26] S. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, Journal of network and computer applications 30 (1) (2007) 114–132.

[27] A. Chandrasekhar, K. Raghuveer, An effective technique for intrusion detection using neuro-fuzzy and radial svm classifier, in: Computer Networks & Communications (NetCom), Vol. 131, Springer, 2013, pp. 499–507.

[28] A. N. Toosi, M. Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers, Computer communications 30 (10) (2007) 2201–2212.

[29] S. Lu, Measuring dependence via mutual information, 2011.

[30] G. Grimmett, D. Stirzaker, Probability and random processes, Oxford Univ Press, 1992, Vol. 2, Oxford Univ Press, 1992.

[31] J. Beauquier, Y. Hu, Intrusion detection based on distance combination, International Journal of Computer Science 2 (3) (2008) 178–186.

[32] A. Jamdagni, Z. Tan, P. Nanda, X. He, R. P. Liu, Intrusion detection using gsad model for http traffic on web services, in: International Wireless Communications and Mobile Computing Conference, ACM, 2010, pp. 1193–1197.

[33] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, W. Jia, W.-c. Yeh, A two-tier system for web attack detection using linear discriminant method, in: Information and Communications Security, Vol. 6476, Springer, 2010, pp. 459–471.

[34] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, Triangle-area-based multivariate correlation analysis for effective denial-of-service attack detection, in: International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2012, pp. 33–40.

[35] T. M. Cover, J. A. Thomas, Elements of information theory, John Wiley & Sons, 2012.

[36] M. S. Roulston, Estimating the errors on measured entropy and mutual information, Physica D: Nonlinear Phenomena 125 (3) (1999) 285–294.

[37] C. E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Computing and Communications Review 5 (1) (2001) 3–55.

[38] K. Das, J. Schneider, Detecting anomalous records in categorical datasets, in: International conference on Knowledge discovery and data mining, ACM, 2007, pp. 220–229.

[39] K. Das, J. Schneider, D. B. Neill, Anomaly pattern detection in categorical datasets, in: International conference on Knowledge discovery and data mining, ACM, 2008, pp. 169–176.

[40] Y. Kopylova, D. A. Buell, C.-T. Huang, J. Janies, Mutual information applied to anomaly detection, Journal of Communications and Networks 10 (1) (2008) 89–97.

[41] Q. Wang, Y. Shen, J. Q. Zhang, A nonlinear correlation measure for multivariable data set, Physica D: Nonlinear Phenomena 200 (3) (2005) 287–295.

[42] S. Cang, H. Yu, Mutual information based input feature selection for classification problems, Decision Support Systems 54 (1) (2012) 691–698.

[43] Z. Shen, Q. Wang, Y. Shen, Effects of statistical distribution on nonlinear correlation coefficient, in: Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE, IEEE, 2011, pp. 1–4.

[44] Q. Wang, D. Yu, Y. Shen, An overview of image fusion metrics, in: Instrumentation and Measurement Technology Conference, 2009. I2MTC'09. IEEE, IEEE, 2009, pp. 918–923.

[45] P. Louvieris, N. Clewley, X. Liu, Effects-based feature identification for network intrusion detection, Neurocomputing 121 (2013) 265–273.

[46] G. H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Machine Learning: Proceedings of the Eleventh International Conference, 1994, pp. 121–129.

[47] Y. Peng, Z. Wu, J. Jiang, A novel feature selection approach for biomedical data classification, Journal of Biomedical Informatics 43 (1) (2010) 15–23.

[48] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters 28 (13) (2007) 1825–1844.

[49] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.

[50] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S. C. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognition 48 (2) (2015) 438–446.

[51] K. Fukunaga, Introduction to statistical pattern recognition, Academic press, 2013.

[52] Y.-I. Moon, B. Rajagopalan, U. Lall, Estimation of mutual information using kernel density estimators, Physical Review E 52 (3) (1995) 2318–2321.

[53] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[54] T. W. Chow, D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, IEEE Transactions on Neural Networks 16 (1) (2005) 213–224.

[55] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemometrics and intelligent laboratory systems 80 (2) (2006) 215–226.

[56] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical Review E 69 (6) (2004) 066138.

[57] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (4) (1994) 537–550.

[58] N. Kwak, C.-H. Choi, Input feature selection for classification problems, IEEE Transactions on Neural Networks 13 (1) (2002) 143–159.

[59] P. A. Estévez, M. Tesmer, C. A. Perez, J. M. Zurada, Normalized mutual information feature selection, IEEE Transactions on Neural Networks 20 (2) (2009) 189–201.

[60] N. D. Thang, Y.-K. Lee, An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information, in: International Symposium on Applications and the Internet (SAINT), IEEE, 2010, pp. 395–398.

[61] P. Mitra, C. Murthy, S. K. Pal, Unsupervised feature selection using feature similarity, IEEE transactions on pattern analysis and machine intelligence 24 (3) (2002) 301–312.

[62] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in neural information processing systems, 2005, pp. 507–514.

[63] L. Koc, T. A. Mazzuchi, S. Sarkani, A network intrusion detection system based on a hidden naïve bayes multiclass classifier, Expert Systems with Applications 39 (18) (2012) 13492–13500.

[64] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. Liu, A system for denial-of-service attack detection based on multivariate correlation analysis, IEEE Transactions on Parallel and Distributed Systems 25 (2) (2014) 447–456.

[65] G. Creech, J. Hu, A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns, IEEE Transactions on Computers 63 (4) (2013) 807–819.

[66] B. Pfahringer, Winning the kdd99 classification cup: Bagged boosting, SIGKDD Explorations 1 (2) (2000) 65–66.

[67] I. Levin, Kdd-99 classifier learning contest: Llsoft's results overview, SIGKDD explorations 1 (2) (2000) 67–75.

[68] D. S. Kim, J. S. Park, Network-based intrusion detection with support vector machines, in: Information Networking, Vol. 2662, Springer, 2003, pp. 747–756.

[69] W. Xuren, H. Famei, X. Rongsheng, Modeling intrusion detection system by discovering association rule in rough set theory framework, in: CIMCA-IAWTIC, IEEE, 2006, pp. 24–24.

[70] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, C. D. Perkasa, A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert systems with Applications 38 (1) (2011) 306–313.

[71] M. Tavallaee, E. Bagheri, W. Lu, A.-A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, 2009, pp. 1–6.

[72] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, K. Nakao, Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation, in: Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, ACM, 2011, pp. 29–36.

[73] G. Grimmett, D. Stirzaker, Probability and random processes, oxford univ press, 2001.

[74] A. H. Bhat, S. Patra, D. Jena, Machine learning approach for intrusion detection on cloud virtual machines, International Journal of Application or Innovation in Engineering & Management (IJAIEM) 2 (6) (2013) 56–66.

[75] E. de la Hoz, A. Ortiz, J. Ortega, E. de la Hoz, Network anomaly classification by support vector classifiers ensemble and non-linear projection techniques, in: Hybrid Artificial Intelligent Systems, Vol. 8073, Springer, 2013, pp. 103–111.

[76] M. Panda, A. Abraham, M. R. Patra, Discriminative multinomial naive bayes for network intrusion detection, in: International Conference on Information Assurance and Security (IAS), IEEE, 2010, pp. 5–10.

[77] W. H. Press, Numerical recipes in Fortran 77: the art of scientific computing, Vol. 1, Cambridge university press, 1992.

[78] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al., A practical guide to support vector classification (2003).

[79] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (3) (1999) 293–300.

[80] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (2) (2012) 513–529.

[81] R. Rifkin, A. Klautau, In defense of one-vs-all classification, The Journal of Machine Learning Research 5 (2004) 101–141.

[82] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, P. K. Chan, Cost-based modeling for fraud and intrusion detection: Results from the jam project, in: DARPA Information Survivability Conference and Exposition, Vol. 2, IEEE, 2000, pp. 130–144.

[83] R. Chitrakar, C. Huang, Selection of candidate support vectors in incremental svm for network intrusion detection, Computers & Security 45 (2014) 231–241.

[84] W. B. Croft, D. Metzler, T. Strohman, Search engines: Information retrieval in practice, Addison-Wesley Reading, 2010.

[85] S. Mukkamala, A. H. Sung, Significant feature selection using computational intelligent techniques for intrusion detection, in: Advanced Methods for Knowledge Discovery from Complex Data, Springer, 2005, pp. 285–306.

[86] S. Chebrolu, A. Abraham, J. P. Thomas, Feature deduction and ensemble design of intrusion detection systems, Computers & Security 24 (4) (2005) 295–307.

[87] Y. Chen, A. Abraham, B. Yang, Feature selection and classification flexible neural tree, Neurocomputing 70 (1) (2006) 305–313.

[88] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications 35 (4) (2008) 1817–1824.

[89] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach, Applied soft computing 8 (4) (2008) 1505–1512.

[90] P. Gogoi, M. H. Bhuyan, D. Bhattacharyya, J. K. Kalita, Packet and flow based network intrusion dataset, in: Contemporary Computing, Vol. 306, Springer, 2012, pp. 322–334.

[91] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, A. E. Hassanien, Hybrid intelligent intrusion detection scheme, in: Soft Computing in Industrial Applications, Vol. 96, Springer, 2011, pp. 293–303.

[92] H. F. Eid, M. A. Salama, A. E. Hassanien, T.-h. Kim, Bi-layer behavioral-based feature selection approach for network intrusion classification, in: Security Technology, Vol. 259, Springer, 2011, pp. 195–203.

[93] S. Mukherjee, N. Sharma, Intrusion detection using naive bayes classifier with feature reduction, Procedia Technology 4 (2012) 119–128.

[94] H. F. Eid, A. E. Hassanien, T.-h. Kim, S. Banerjee, Linear correlation-based feature selection for network intrusion detection model, in: Advances in Security of Information and Communication Networks, Vol. 381, Springer, 2013, pp. 240–248.

[95] M. M. Abd-Eldayem, A proposed http service based ids, Egyptian Informatics Journal 15 (1) (2014) 13–24.

[96] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Systems with Applications 41 (4) (2014) 1690–1700.

[97] R. Agarwal, M. V. Joshiy, Pnrule: A new framework for learning classi er models in data mining (a case-study in network intrusion detection), Citeseer2000.

[98] C.-H. Tsang, S. Kwong, H. Wang, Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, Pattern Recognition 40 (9) (2007) 2373–2391.

[99] M. Aghamohammadi, M. Analoui, A comparison of support vector machine and multi-level support vector machine on intrusion detection, World of Computer Science and Information Technology Journal 2 (7) (2012) 215–219.

[100] Q. Liu, J. Yin, V. C. Leung, J.-H. Zhai, Z. Cai, J. Lin, Applying a new localized generalization error model to design neural networks trained with extreme learning machine, Neural Computing and Applications (2014) 1–8.

[101] Y. Bouzida, F. Cuppens, Neural networks vs. decision trees for intrusion detection, in: IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), Tuebingen, Germany, 2006, pp. 28–29.

[102] A. W. Whitney, A direct method of nonparametric measurement selection, IEEE Transactions on Computers 100 (9) (1971) 1100–1103.

[103] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern recognition letters 15 (11) (1994) 1119–1125.

[104] S. Nakariyakul, D. P. Casasent, An improvement on floating search algorithms for feature subset selection, Pattern Recognition 42 (9) (2009) 1932–1940.

[105] D. Zhang, S. Chen, Z.-H. Zhou, Constraint score: A new filter method for feature selection with pairwise constraints, Pattern Recognition 41 (5) (2008) 1440–1451.

[106] D. Zhang, J. He, Y. Zhao, Z. Luo, M. Du, Global plus local: A complete framework for feature extraction and recognition, Pattern Recognition 47 (3) (2014) 1433–1442.

[107] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering., in: NIPS, Vol. 14, 2001, pp. 585–591.

[108] X. Niyogi, Locality preserving projections, in: Neural information processing systems, Vol. 16, MIT, 2004, p. 153.

[109] F. R. Chung, Spectral graph theory, Vol. 92, American Mathematical Soc., 1997.

[110] D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz, {UCI} repository of machine learning databases, univ. california at irvine, 1998 [online]. available https://archive.ics.uci.edu/ml/datasets.html.

[111] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.