A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks

Kamarudin, M. H., Maple, C., Watson, T. & Safa, N. S.

Original citation:
Kamarudin, MH, Maple, C, Watson, T & Safa, NS 2017, 'A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks' Security and Communication Networks, vol. 2017, 2539034.
https://dx.doi.org/10.1155/2017/2539034

DOI 10.1155/2017/2539034
ISSN 1939-0114
ESSN 1939-0122

Publisher: Hindawi Publishing Corporation: Wiley Hindawi Partnership

Hindawi
Security and Communication Networks
Volume 2017, Article ID 2539034, 18 pages
https://doi.org/10.1155/2017/2539034

Research Article

A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks

Muhammad Hilmi Kamarudin, Carsten Maple, Tim Watson, and Nader Sohrabi Safa

Cyber Security Centre, Warwick Manufacturing Group, University of Warwick, Coventry CV47AL, UK

Correspondence should be addressed to Muhammad Hilmi Kamarudin; m.h.b.kamarudin@warwick.ac.uk

Received 7 June 2017; Revised 19 August 2017; Accepted 5 September 2017; Published 7 November 2017

Academic Editor: Ángel Martín Del Rey

The global usage of more sophisticated web-based application systems is obviously growing very rapidly. Major usage includes the storing and transporting of sensitive data over the Internet. The growth has consequently opened up a serious need for more secured network and application security protection devices. Security experts normally equip their databases with a large number of signatures to help in the detection of known web-based threats. In reality, it is almost impossible to keep updating the database with the newly identified web vulnerabilities. As such, new attacks are invisible. This research presents a novel approach of Intrusion Detection System (IDS) in detecting unknown attacks on web servers using the Unified Intrusion Anomaly Detection (UIAD) approach. The unified approach consists of three components (preprocessing, statistical analysis, and classification). Initially, the process starts with the removal of irrelevant and redundant features using a novel hybrid feature selection method. Thereafter, the process continues with the application of a statistical approach to identifying traffic abnormality. We performed Relative Percentage Ratio (RPR) coupled with Euclidean Distance Analysis (EDA) and the Chebyshev Inequality Theorem (CIT) to calculate the normality score and generate a finest threshold. Finally, Logitboost (LB) is employed alongside Random Forest (RF) as a weak classifier, with the aim of minimising the final false alarm rate. The experiment has demonstrated that our approach has successfully identified unknown attacks with greater than a 95% detection rate and less than a 1% false alarm rate for both the DARPA 1999 and the ISCX 2012 datasets.

1. Introduction

The continuous growth of Internet usage, development of speedier Internet technology, and availability of massive sensitive information have caused servers to be the primary target of malicious attack. Lately, web-based applications and web servers have become popular targets as most network communication serves client-server enquiry needs. These web applications are often accessible through ports that are open through firewalls [1]. Although the Internet provides convenient real-time information services to the public, the potential threats to confidentiality, integrity, and availability (CIA) need to be addressed more effectively and permanently [2]. To fortify the security aspects of web-based servers and systems, Intrusion Detection Systems (IDSs) can be used as a complementary device to many existing security appliances such as password authentication, firewalls, access control, and vulnerability assessments.

The *Intrusion Detection System* (IDS) is an application system or a device that identifies hostile activities or policy violation activities within a network. IDSs have been widely used in recent years as one of the network security components. They play an active role in network surveillance, as well as functioning as a network security guard. IDSs function to capture and analyse traffic movement and precipitate an alarm when there is an intrusive action detected. The alarm is set to alert the security analyst to take the necessary action. In general, IDSs can be designed either as a network-based IDS (NIDS) or as a host-based IDS (HIDS) [3] to recognise signs of intrusion. The design is based on the placement of the IDS either to capture traffic for whole network or only for a specific host [4]. In NIDSs, the IDS is normally installed before and after the firewall to capture traffic for whole network segment. With respect to the HIDS, the IDS focuses on a specific host to examine packets, logs, and system calls.

Such being the case, the HIDS is more suitable in identifying internal attacks compared to the NIDS [5].

According to [6] there are two types of IDS: The *Signature Detection System* (SDS) and the *Anomaly Detection System* (ADS). In SDS, a set of previously defined rules are stored inside databases specifically used to identify known attacks. In view that SDS technique relies on consistent signature updates, it is unable to detect unknown or new attacks [7]. Consequently, such attacks could pass through the system undetected. On the other hand, the ADS approach is based on analysis of normal behaviour traffic as the baseline of general usage patterns. Fundamentally, ADS is based on the assumption that any traffic that deviates from normal patterns will be identified as malicious traffic [8]. The main advantage of this approach is its ability to identify new or unknown attacks. In spite of having such advantages, ADS are overly keen to trigger massive false detection [9]. A false detection occurs when the system misclassifies legitimate traffic as malicious traffic and vice versa. The key factor in ADS is developing a system that could produce high detection accuracy while maintaining low false detection rates.

Therefore, this paper presents a novel Unified Intrusion Anomaly Detection (UIAD) that consists of three components (preprocessing, statistical analysis, and classification). The study provides contributions through a new set of techniques using 2-stage detection which aims to improve the outlier detection rate and minimise the false alarm rate in ADS environments. Initially, we performed hybrid feature selection (HFS) to filter out the irrelevant and redundant features. Secondly, the first-stage detection begins with statistical approaches, where the methods are further divided into two phases: the learning phase and the detection phase. Meanwhile, in the second-stage detection, the data mining approach is employed, and in particular it uses ensemble learning classification to improve the true detection rate (True Positive and True Negative) and the misclassification rate (False Positive and False Negative) that have first been detected in the first stage. Finally, we implemented the Logitboost algorithm as a metaclassifier with RF as a base classifier. The result has demonstrated a significant improvement regarding attack detection accuracy and a reduction in the false alarm rate for both the DARPA 1999 and ISCX 2012 datasets.

The rest of this paper is organised as follows. Sections 2 and 3 review the related work and the datasets used by this study, while the proposed approaches are explained in Section 4. The experimental results are presented in Section 5. Section 6 concludes and outlines future work.

2. Related Work

In this section, we discuss the related works on IDSs and the existing work in the areas of feature selection, statistical analysis, data mining algorithms, and web-attack traffic.

2.1. Feature Selection. Feature selection is a foundation of machine learning and has been studied for many years [22]. It is a process of discovering the most prominent features for the learning algorithm in the sense that the most useful data is analysed for better future projection. Therefore, it is imperative to extract the redundant or irrelevant features to provide excellent discriminative models for every classifier. As the effectiveness of the selected algorithm is highly dependent on the feature selected, it is also crucial to choose the most significant features that could contribute to maximising the classification performances. Selecting the feature selection algorithm often requires expert knowledge, as it is not a straightforward task to identify a good set of features.

Currently, the two general methods used in this field are the filter and wrapper [23] approaches. Filter-based subset evaluation (FBSE) was introduced simply to overcome the redundant feature issue inside filter-ranking [24]. It examines the whole subset in a multivariate way. It selects the relevant features and explores the degree of relationship between them. In addition, FBSE is heuristic-based and involves probabilities and statistical measures to search for and evaluate the usefulness of all identified features. On the other hand, the wrapper-based subset evaluation (WBSE) uses a classifier to estimate the worthiness of feature subset. Usually, WBSE has better predictive accuracy compared to filters. This is because the selection approach is optimised when evaluating each feature subset with a particular classification algorithm.

Conversely, most of the time wrappers use a classification algorithm to evaluate each set of features. This has made it excessively expensive to execute. Moreover, when dealing with a large database that consists of many features, [25] the wrapper can become uncontrollable. Wrappers are also highly associated with the classifier's algorithm and that makes it more difficult when shifting from one classifier to another because the selection process needs total reinitiation. Unlike filters, the selection criteria of features use distance measures and correlation functions [26]. It does not require reexecution for different learning classifiers. As such, its execution is much faster than wrappers. Filters are suitable in large database environments that contain many features. Researchers have often used the filter method as an alternative to the wrapper method, since the latter is expensive and time-consuming to run.

2.2. Statistics-Based Approaches. Statistical methods in IDSs were first introduced by [27]. The detection approach primarily relies on a collection of data history to create a normal profile of behaviour. In this approach, only benign traffic data collected over a period of time is utilised to detect intrusion [27]. Some researchers have proposed a statistical model in more specific areas such as Packet Header Anomaly Detection (PHAD). In PHAD packet characteristics and behaviours are used to recognise abnormal patterns. PHAD uses statistical measurement from activity history [28] to construct a normal profile. A set of traffic that deviates from the normal profile and behaves abnormally would be identified as an intruder by this method. Instead of using IP addresses and port numbers, PHAD uses all information inside a packet header [28]. The 33 attributes in a packet header represent the information of 3 layers in the OSI 7-layer model, which are the data link, network, and transport layers. The information in the attributes is used to measure the

probability of each packet being normal or tending towards abnormal behaviour. The anomaly score is awarded when there is any dissimilarity detected between training data and the testing data. Finally, the sum anomaly score of each packet is totalled up and flagged as anomalous if the score surpasses the preset threshold.

In contrast to conventional PHAD systems, [29] proposed the Protocol-based Packet Header Anomaly Detection (PbPHAD) in two different environments: network-based and host-based. The proposed method used three main protocols: the *transmission control protocol* (TCP), *user datagram protocol* (UDP), and *Internet control messaging protocol* (ICMP) to construct a normal profile that contains normal behaviour. Similar to the traditional PHAD system, this approach uses all 33 packet header attributes to produce an anomaly score. The score will individually rate the degree of incoming traffic. In spite of surpassing the results from PHAD and DARPA best system [30] with a 57.83% detection rate, there is still room for further improvement.

To identify whether malicious packets exist inside Telnet traffic, [6] has proposed the Lightweight Network Intrusion Detection System (LNID). In LNID, benign behaviour extracted from training data is used to construct a normal profile. Additionally, the normal profile is used as the indicator to compute an anomaly score. The anomaly score was given during a matching process between testing and training data. The packets are flagged as malicious when the score surpassed the preset threshold. Insignificant features from training data are removed during the preprocessing phase to reduce computational cost. Although the scoring approach in LNID has increased the detection rate for U2R and R2L to 86.4%, it still has the opportunity for further improvement. The test has recorded nearly 14% of undetected attacks by singly using anomaly scores to determine the threshold without considering effective features as additional input. Profile generation has attracted [15] to propose catastrophe and equilibrium surface theory to extract common behaviours that exist within the network. The standard equilibrium surface is used to indicate the change of packet behaviour, which makes it suitable for inspecting incoming traffic. Despite the fact that the evaluation of true positives increased to slightly over 86% for Telnet traffic, the real challenge is to get the best detection rate together with the lowest false alarm rate.

2.3. Data Mining Based Approaches. Data mining is the technique of discovering systematic data relationships and determining the fundamentals of data information [7]. Data mining is divided into two broad categories: unsupervised and supervised approaches. Clustering and classification are examples of unsupervised and supervised algorithms, respectively. In clustering, the group of objects are based on characteristic data points, where every single data point in a cluster is similar to those within its cluster but is dissimilar to those in a different cluster. It works by grouping similar data into one or more clusters to ease abnormality identification. However, this approach would potentially increase the false alarm rate. In view of the fact that IDS performance is highly dependent on its achieving a low false alarm rate, its capabilities would be downgraded if it continuously generated a high false alarm rate. For that reason, classification is the better approach in classifying (i.e., benign or anomalous) data, especially in reducing the false alarm rate. Classification is the supervised approach that has the capability to differentiate unusual data patterns, thus making it suitable for the identification of new attack patterns [31]. Furthermore, classification has been widely used due to its strong reliability in identifying normal structure accurately, which contributes towards reducing false detection [32].

The ensemble technique in classification has attracted researchers to perform a combination of several classifiers which aim to obtain better prediction on accuracy performance [33]. The ensemble method is divided into 3 main approaches: (i) bagging, (ii) stack generalisation, and (iii) boosting. Bagging often referred to as "bootstrap aggregating" functions to improve detection accuracy by fusing the outputs of learned classifiers into a single prediction. For instance, the RF algorithm achieves high classification accuracy by fusing random decision trees using the bagging technique. Stack generalisation, or stacking, basically involves the combination of predictions from several learning algorithms. The prediction output from base-level classifiers is used to achieve high generalisation accuracy.

Boosting is mainly used to boost weak classifiers or weak learners to achieve a higher accuracy classifier. In other words, boosting can be considered a metalearning algorithm. The incorrectly classified instances from the previous model are used to build an ensemble. Weak classifiers such as decision stumps that are based on a decision tree with a root node and two leaf nodes are usually used in the boosting technique [34]. Adaboost (Adaptive boosting) is the most popular boosting algorithm which was first introduced by [35]. The high accuracy achieved by using this algorithm has attracted researchers [36–38] to employ this method in IDSs.

In [36], the author has proposed Adaboost, with a decision stump as a weak classifier. The noise and outliers existing inside the dataset are initially removed by training the full data. The sample data that contained high weight is considered as noise and as containing outliers. Although the detection rate achieved was almost 92%, the false alarm rate was still at 8.9%. Similarly in [20], the authors had proposed CAGE (Cellular Genetic Programing) that used the evolve combination function present in ensemble approaches. The approach was tested on the ISCX dataset and achieved a 91.37% attack detection rate. Although the approach achieved a high detection rate, the recorded high false alarm rate constitutes a limit to the system's capability.

In choosing the right weak classifier for Adaboost, [37] has compared four classifiers NNge (Nonnested generalised exemplars), JRip (Extended Repeated Incremental Pruning), RIDOR (Ripple-Down Rule), and Decision Tables as a base classifier for Adaboost. The proposed combination of Adaboost with NNge received the highest detection rate in detecting U2R and R2L types of attack while a combination of Adaboost with Decision Tables was found to be efficient in detecting DoS attack. Paper [38] has proposed a similar concept to [36]. The author has tested a Naïve Bayes algorithm to be used as weak classifier. Although the proposed algorithm

could achieve a 100% detection rate for DoS attacks, the overall performance (84% detection rate with 4.2% false alarm rate) is still much lower compared to [36].

The introduction of the *logistic-regression* (Logitboost) algorithm [39] as an alternative solution is to address the drawback of Adaboost in handling noise and outliers. The Logitboost algorithm uses a binomial log-likelihood that changes the loss function linearly. In contrast, Adaboost uses an exponential loss function that changes exponentially with the classification error. This is the reason why Logitboost turns out to be less sensitive to outliers and noise. To the best of our knowledge, no research to date has investigated the performance of the Logitboost algorithm in the field of ADS environment.

2.4. Attack on Web Traffic. The exposure of vulnerable web-based applications and their related sensitive information in the Internet environment has promoted network security to an area of major concern. This is because incidents of attack are getting more frequent and aggressive, causing serious damage to the targeted web-based information system. As such, it is not surprising that more researchers are involved in this field. Paper [40] has proposed a learning based approach to secure web servers by focusing on detecting SQL and Xpath injection attacks. The detection is based on an input query where the attacker usually adds extra conditions to the original SQL commands. The proposed methods examine the structure and type of inputs as well as the outputs of the operation existent in the XSD file. After the necessary information is collected, the workload generator is used to inspect the set of data accessing the SQL/Xpath presented in the source code. In detection mode, the SQL query is compared with the normal SQL query that contains zero attacks stored in the lookup map. If the SQL query is not found, the execution will stop processing the query to avoid probable hazardous requests. This approach is capable of alerting developers and service administrators to stop the XPath/SQL injection before the system is harmed.

The research proposed in [41] assumed that the attack patterns commonly have a level of complexity that exceeds normal access requests. This complexity is used as a benchmark in detecting attacks. The recorded request log is inspected with Shannon entropy analysis, which is used to calculate the complexity level. In defining entropy level, normal log requests in training sets are used as benchmarks of a legitimate profile. The boundaries (threshold) in detection are measured using average and standard deviation of the period for each entropy. Log requests that surpass the predefined complexity threshold are flagged as potential intrusions. Although the proposed attack detection approach is able to detect attacks at a satisfactory rate, the false detection rate has room for further improvement.

On the other hand, the work done by [42] has considered the analysis on HTTP log requests. The normal HTTP log requests have been used as training sets that could describe the model of normal user behaviour. This approach is similar to work from [43], where they make use of query in detecting SQL attacks. Initially, the query trees would be converted into dimensional vectors for feature extraction and feature transformation. The work has been carried out using the data mining technique and utilise the *Support Vector Machines* (SVM) algorithm for classification purposes. The result has demonstrated conspicuous performance improvement in terms of computational time reduction and attack detection accuracy rates. Although the detection rate presents a significant improvement, the proposed methods require a readable payload to extract the http request.

Traditionally, IDS works with the principle of "deep packet inspection" where the packet payloads are inspected to look for the presence of malicious activities. As the usage of network communications gets more frequent, the demands for more secured communication using cryptography also increase. In encrypted traffic environments, *secure sockets layer* (SSL), *wired equivalent privacy* WEP, or *Internet protocol security* (IPsec) protocols are utilised to offer better privacy and confidentiality. Previous work in detecting web-based attacks mainly focused on investigating the log/payload content [44, 45]. In view that the traffic is encrypted, payload (log) is unavailable as the content is indecipherable. Unlike payload, the information set of the packet header is still accessible for extrication. Packet headers were used in this research due to the availability of information even in encrypted traffic situations. Thus, this approach is applicable in detecting malicious attacks within the encrypted network traffic.

3. Dataset Description

The proposed method was experimented using two different datasets: the DARPA 1999 [30] and the ISCX 2012 [46] datasets. We made use of a publicly available labelled dataset simply to avoid the problems described in [47] with recorded traffic from the real environment. Both datasets are available online and have been comprehensively used as a standard benchmark by many researchers in this field, for example, by [6, 15, 16, 48]. The DARPA 1999 dataset is traditional and commonly used in this field. Basically, the dataset is an improved version of the Defence Advanced Research Projects Agency (DARPA) 1998 initiative updated with additional types of attack. In contrast, the ISCX 2012 dataset is a modern updated dataset, which is claimed to have rectified the weaknesses identified in DARPA 1999.

3.1. DARPA 1999. MIT Lincoln Lab has provided a publicly available dataset called DARPA 1999. The dataset consists of traffic data spanning a total of 5 weeks, with 3 weeks of training and 2 weeks of testing data. In view of its multiformat datasets, we choose tcpdump since it contains comprehensive TCP/IP information that is good for traffic analysis. In training data, traffic from weeks 2 and 3 was defined as benign traffic as it is free from attack. Thus, it is suitable to use the data to train ADS. For testing data, weeks 4 and 5 contain attacks that were generated in the middle of benign traffic. The distributions of the attacks are different for week 4 and for week 5. In week 5, the data contains more attacks that were not present in week 4. The different attack distribution is an

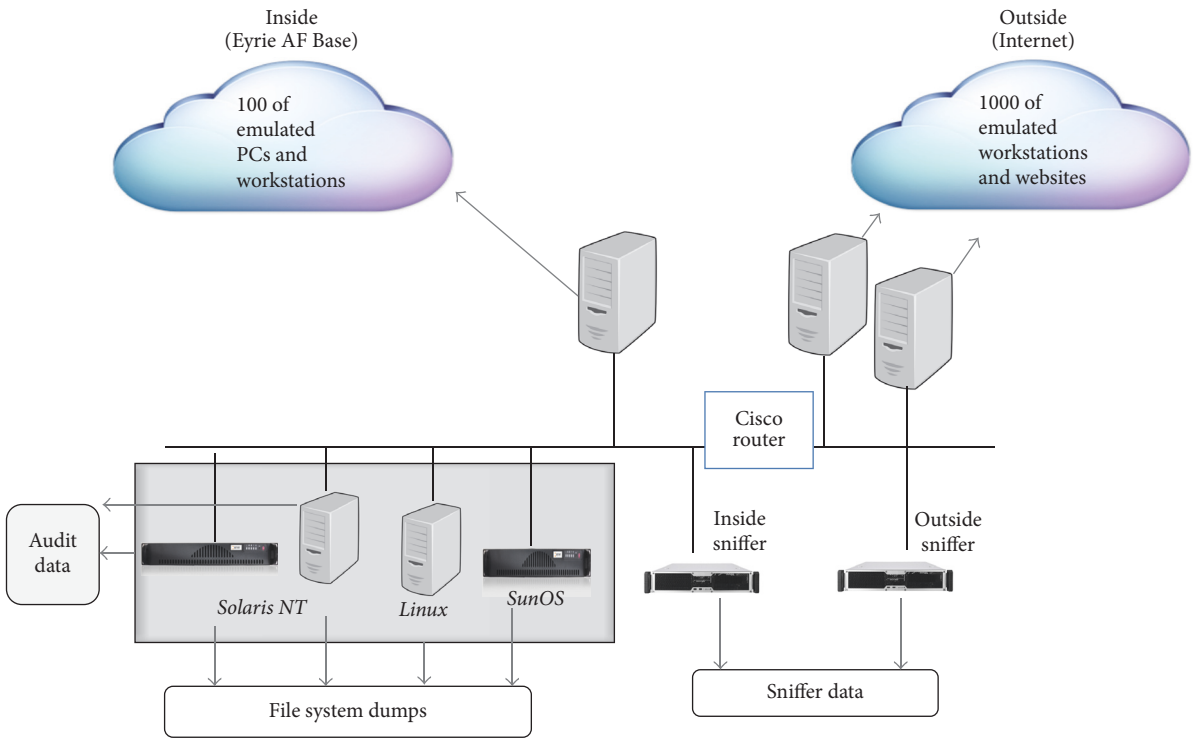

Figure 1: Diagram of 1999 test bed simulation [30].

opportunity for researchers to seek methods that can be used to detect new or novel attacks.

Figure 1 shows the dataset generation simulation based on a scripting technique generating live benign and attack traffic. The scenario is equivalent to flowing traffic from the internal Eyrie Air Force Base to the Internet at large. The test bed generates rich background traffic to simulate the initialisation of traffic, as if the traffic was initiated by thousands of hosts from hundreds of users. All attacks were set to automatically launch against victim machines (UNIX OS) and the external host's router. The sensor known as "sniffer" was placed within the internal and external network to capture all traffic broadcasted through the network.

3.2. ISCX 2012. This dataset was generated by [46] from the University of Brunswick (UNB) and aimed to address issues in other existing datasets such as DARPA, CAIDA, and DEFCON. The 7-day simulation dataset consists of 3 days of attack-free traffic and 4 days of mixed benign and malign traffic. The distribution model profile concept is the basis of the dataset effectiveness in realism, evaluation, malicious activity, and capabilities. Numerous multiphase attack events were induced to create the anomaly trace to the dataset such as HTTP Denial of Service (DoS), Botnet, Distributed Denial of Service (DDoS), and Brute Force SSH. The simulation was created to simulate and mimic user behaviour activity. Profile-based user behaviour was created by executing a user-profile that synthetically generates at random synchronized times. The dataset came with labelled traffic that could assist the researcher for testing, comparison, and evaluation purposes.

Figure 2 shows the ISCX 2012 test bed network that contains 21 interconnected Windows workstations. Those workstations were equipped with Windows operating systems as a platform to launch attacks against the test bed environment. Out of 21, 17 workstations were installed with Windows XP SP1, 2 with SP2, 1 with SP3, and the rest with Windows 7. The network architecture divides the workstation into four distinct LANs. This configuration was expected to represent a real connectivity network environment. The servers located at the fifth LAN provide web, email, DNS, and Network Address Translation (NAT) services.

The NAT server (192.168.5.124) was placed at the entry point of the network so that the firewall would only allow authorised access. The primary main server (192.168.5.122) was accountable for email services, delivering website and performing as an internal name resolver. The secondary server (192.168.5.123) was made responsible for handling internal ASP.NET applications that sit on Windows Server 2003 machines. Both main and NAT servers were run on Linux operating systems and configured with Ubuntu 10.04. Our experiment was focused on the specific host server addresses DARPA (172.016.114.050) and ISCX (192.168.5.122). These two hosts were chosen due to their having the highest attack traffic content.

4. Methodology

In this research, our anomaly detection approach consists in three parts: preprocessing (hybrid feature selection), statistical analysis (benign behaviour analysis), and data mining (boosting classification algorithm).

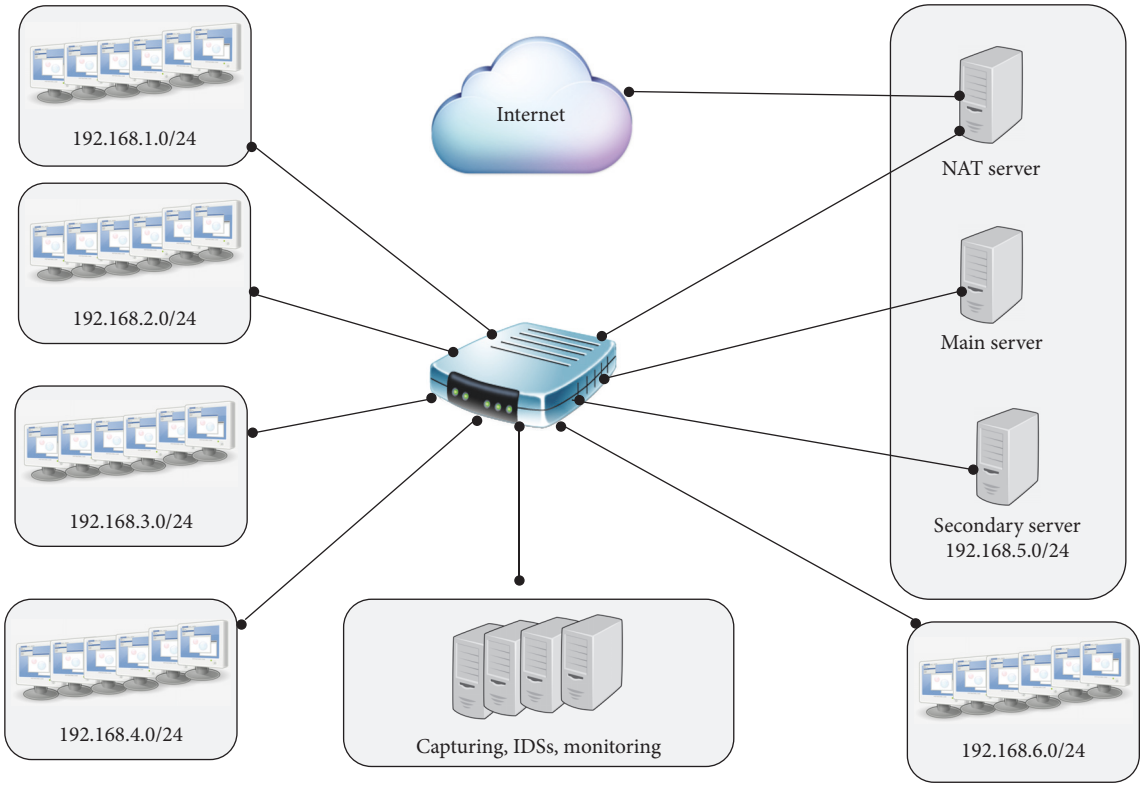

Figure 2: ISCX 2012 test bed network architecture [46].

4.1. Preprocessing. In the preprocessing step, we adopted our previous HFS [49] approach to leverage the strengths of both the filter and wrapper approaches. In addition, the proposed filter-based subset evaluation (FBSE) was utilised to resolve the drawback in filter-ranking where redundant features exist.

Figure 3 shows the process flows for building HFS, which can be classified into 3 phases as follows.

In Stage 1, the process starts with the filter-subset evaluation. It processes the original features M and produces a new set L of reduced features, where $L \subseteq M$. We proposed the Correlation Feature Selection (CFS) approach due to its robustness in removing redundant and irrelevant features. This approach prevails to overcome the existence of redundant features, as in CFS the relationship between features is measured as in (1). In addition, in feature ranking, the reduced features are usually defined without the need to perform further examination (information gain, gain ration). The CFS is an intelligible filter algorithm that evaluates subsets of features based on a heuristic evaluation function. The evaluation is based on the hypothesis *"A good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other"* [25].

$$Ms = \frac{k\,\overline{\text{rcf}}}{\sqrt{k + k\,(k-1)\,\overline{\text{rff}}}}. \tag{1}$$

Equation (1) shows how the merit, M, is used to select subset s containing k number of features. Both redundant and irrelevant features are determined by the $\overline{\text{rcf}}$, which represents the mean of the relationship of each feature to its class while the $\overline{\text{rff}}$ is the mean of the relationship among the features. The exhaustive search is not suitable in large datasets [25] due to its high complexity. As such, we used heuristic search techniques and chose a genetic algorithm as the search function. This was because our experiment reveals that the genetic algorithm gives a global optimum solution and is more robust compared to the best-first and greedy methods. Furthermore, at this stage it is crucial to help to truncate the computational effort using the wrapper approach as it only deals with a reduced set of features compared to the original set of features.

In Stage 2, the reduced feature set L gathered from the FBSE was further processed by WBSE to produce the final set of optimal features K, where $K \subseteq L \subseteq M$. The proposed filter and wrapper hybridisation approaches would leverage both of their strengths to produce a much better result in terms of accuracy, false alarm rate, and fewer redundant and irrelevant features. This was due to the fact that the filter approach could not find the best available subset, as it is less dependent on the classifier. On the other hand, the wrapper approach is proven to be more effective and produced better accuracy. Nevertheless, it is computationally expensive when dealing with a large dataset. Thus, by leveraging the strengths of both methods, we had combined both methods together to form a hybrid feature selection (HFS) approach. We use the Random Forest (RF) classifier in WBSE to evaluate the selected features using genetic search and produced the final

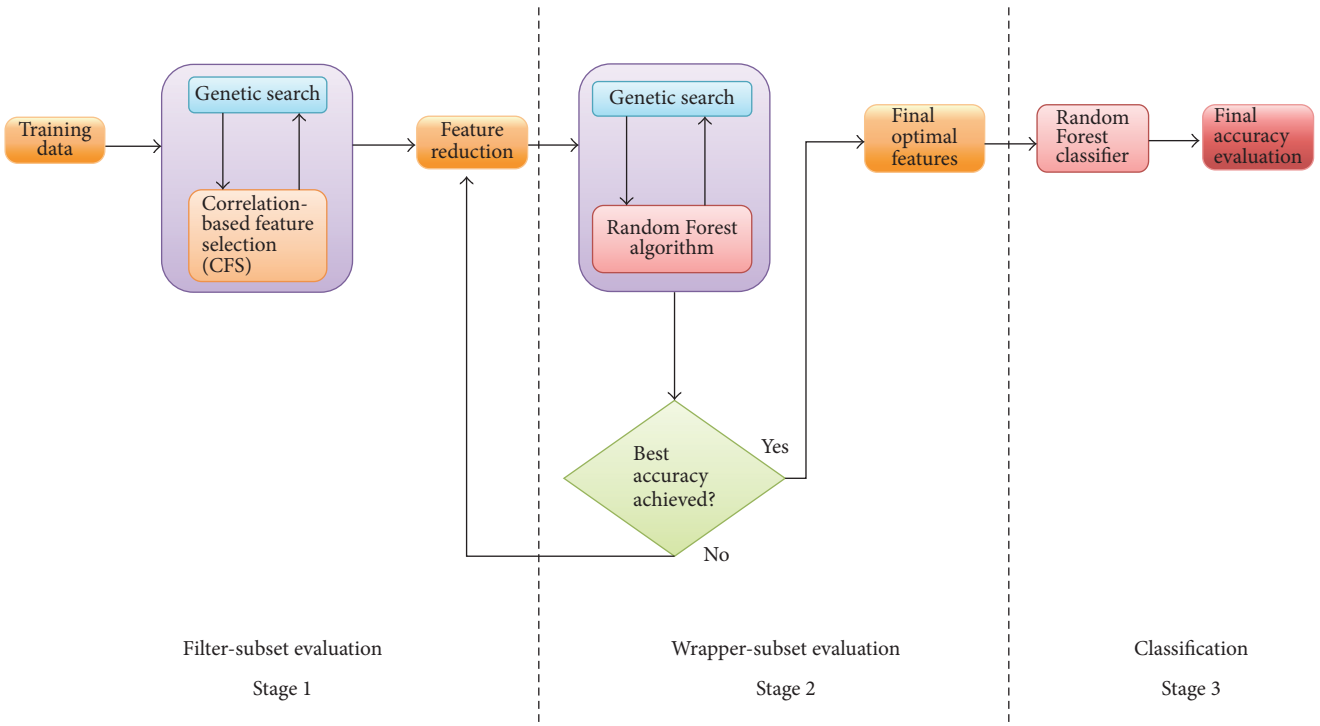

Figure 3: Hybrid feature selection (HFS) design [49].

K feature subset. The search would continue to train a new model for each subset and will stop once the final optimum subset is found.

Stage 3 is called the classification stage. In this stage, the final optimum subset K, produced by WBSE, was tested by the RF classifier with 10-fold cross validation. RF consists of many decision tree classifiers. Each decision tree was constructed from the different original dataset samples. The outputs were chosen based on votes obtained from each tree that indicated the tree's decision concerning the class object. The most votes for the object are from the best individual trees.

The RF algorithm is widely used in data mining techniques for prediction, pattern recognition, and probability estimation, as in [51–53].

Figure 4 presents the general architecture of RF. As RF originates from many decision trees, each tree of RF is grown by a different sample of bootstrap using a decision tree as a weak classifier. The vote is given by each tree to represent the tree's decision towards the class object. The forest will choose the class with the majority vote of all over the trees. Out-of-bag (OOB) error is used as validation during the tree growth. It is described as the average of the classification error connected to each tree T_b using the OOB_b sample. After constructing the forest, a new sample x_i needs to be classified according to following equation:

$$\hat{c}_{\text{rf}}^{B}(x_i) = \text{majority vote}\left\{\hat{c}_b(x_i)\right\}_1^B, \tag{2}$$

where $\hat{c}_b(x_i)$ is the class that is assigned by the tree T_b.

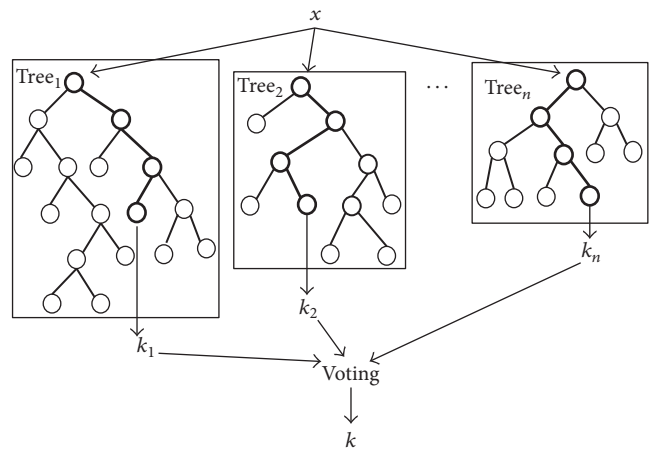

Figure 4: A general architecture of a Random Forest [50].

4.2. Statistical Based Anomaly Detection (First-Stage Detection). Although some work has been done in the past directed towards determining how to detect abnormalities using header traffic, for example, [6, 28], this work does not take into account the influences of packet size while analysing benign and abnormal traffic. In our research, through statistical based anomaly detection (SBAD), we computed the attack detection performance by calculating the traffic normality alongside with standard profile and packet size.

4.2.1. Standard Profile. At bottom, the ADS is based on the analysis of the normal behaviour detection model. The

normal practice model examines the incoming traffic against its standard normal behaviour to divulge any significant irregular patterns. Using benign traffic rather than abnormal behaviour as a profile was found to be more convenient, as the intruder tends to employ certain evasion techniques. In addition, to determine the difference between anomalous and benign traffic, the probability of discrepancy traffic was calculated using statistical techniques which assign an anomaly score function. The basic idea of generating a normal profile was proposed by Mahoney and Chan [28] using a nonstationary model. The model is based on the possibility of the event, depending on the time since it last happened. The study made by Chen et al. [6] concluded that the nonstationary model is not suitable to detecting attacks which occurred on a different time scale. For instance, for 2 *httptunnel* attacks that shared the same traffic content T and T', where T occurred 1 s after the previous attack and T' happened 30 mins after the previous attack, both attacks should share the same anomaly score due to the same packet content. However, the difference on t value of both attacks, one with $T = 1$ and the other with $T' = 1800$, has resulted in different anomaly scores. The different anomaly scores for both packets reflect the gap time that occurred between T and T', resulting in an anomaly score for T' which is 1800 times greater than that for T. As both attacks shared the same content, conveniently they should have the same anomaly score. This approach will give effect when the threshold is set to the certain level, where T might be ignored after the system detected T'. To address these issues, Chen et al. [6] have introduced stationary models that ignored the time dependent scheme.

We adopted the idea of extracting distinct values from attack-free traffic introduced by Mahoney and Chan [28] and stationary models proposed by [6], since such approaches are able to demonstrate traffic characteristics efficiently. Nevertheless, our approach is different, in so far as it does not solely depend on normal profiles to determine malign traffic. Our model can be seen as a unified system, which consists of feature selection, statistical, and data mining approaches. Our research approach is different from [6, 28] in three ways. Firstly, we eliminated superfluous and irrelevant features using our proposed hybrid feature selection methods. Secondly, we used a normal score conjunction with packet size features to produce a better threshold mechanism. In our research, we propose measuring the normal score instead of calculating the anomaly score. The main reason for us calculating the normal score as an alternative to the anomaly score proposed by [6, 28, 29] was because the latter is not sensitive to considering the new value in an attribute. In [6], our observation revealed that benign traffic is more likely to have more novel value than malign traffic. Furthermore, in the real environment, there is more benign traffic compared to malign traffic. Thus, analysing the degree of normal field value in the traffic is appropriate and easier compared to doing so for attack traffic. Thirdly, we proposed 2-stage detection strategy comprise of a statistical approach alongside with Logitboost algorithm with the aim of reducing the overall false detection rate. In our statistical approach, the practice of treating normal traffic behaviour as a standard profile has limited the system's ability to recognise attack behaviour. Thus, by implementing a classification approach using data mining, additional derived features from statistical procedures along with variation samples of malicious traffic could define attack behaviour more precisely. As a result, the detection accuracy and misclassification rate would be greatly improved.

Table 1: Standard profile (DARPA 1999).

k	Features	R_k	N_k	$\log(R_k/N_k) * 100$
1	ethersize	235	53533	*8.36*
2	ethersourcehi	4	53533	*14.63*
3	ethersourcelo	5	53533	*14.29*
4	iplength	36736	53533	*0.58*
5	ipfragid	236	53533	*8.35*
6	ipsource	15	53533	*12.59*
7	tcpsourceport	5134	53533	*3.61*
8	tcpheaderlen	2	53533	*15.69*
9	tcpflag	5	53533	*14.29*
10	tcpwindowsize	382	53533	*7.61*
Total normality score				*100*

Table 2: Standard profile (ISCX 2012).

k	Features	R_k	N_k	$\log(R_k/N_k) * 100$
1	totalSourceBytes	4032	25961	*10.75*
2	sourceTCPFlagDescription	14	25961	*43.46*
3	source	36	25961	*38.00*
4	sourcePort	6739	25961	*7.79*
Total normality score				*100*

Figure 5 shows how the proposed ADS model is divided into two phases. In the learning phase, we created a standard profile as a benchmark to determine benign traffic characteristics. The purpose of creating the profile was to identify and calculate the degree of normality for the incoming web traffic (benign or malign). Normal scores were given for every associate feature that was based on the procedure shown in (3) while a standard profile is given in Tables 1 and 2.

We index attributes as k, where $k = 1, 2, 3, 4, \ldots, n$ and R_k is a distinct accumulation of standard packet characteristics while N_k is the total amount of traffic related to each attribute.

$$\text{Normal Score} = \sum_{k=1}^{n} \frac{R_k}{N_k}, \quad k = 1, 2, 3, \ldots, n. \tag{3}$$

Tables 1 and 2 illustrate the generic model of the standard profile. The R value represents a distinct value for each attribute. We use a log ratio in our model to calculate the score as the R value varies greatly. The normal score is calculated based on distinct values divided by the total number of traffic (R/N). The proportion score is multiplied by 100 to get the percentage values.

In the detection phase, the testing data contains a mixture of benign and malign traffic. As can be seen in Figure 6, the web-based traffic within the testing dataset was matched with the standard profile. We incorporated the scores derived from

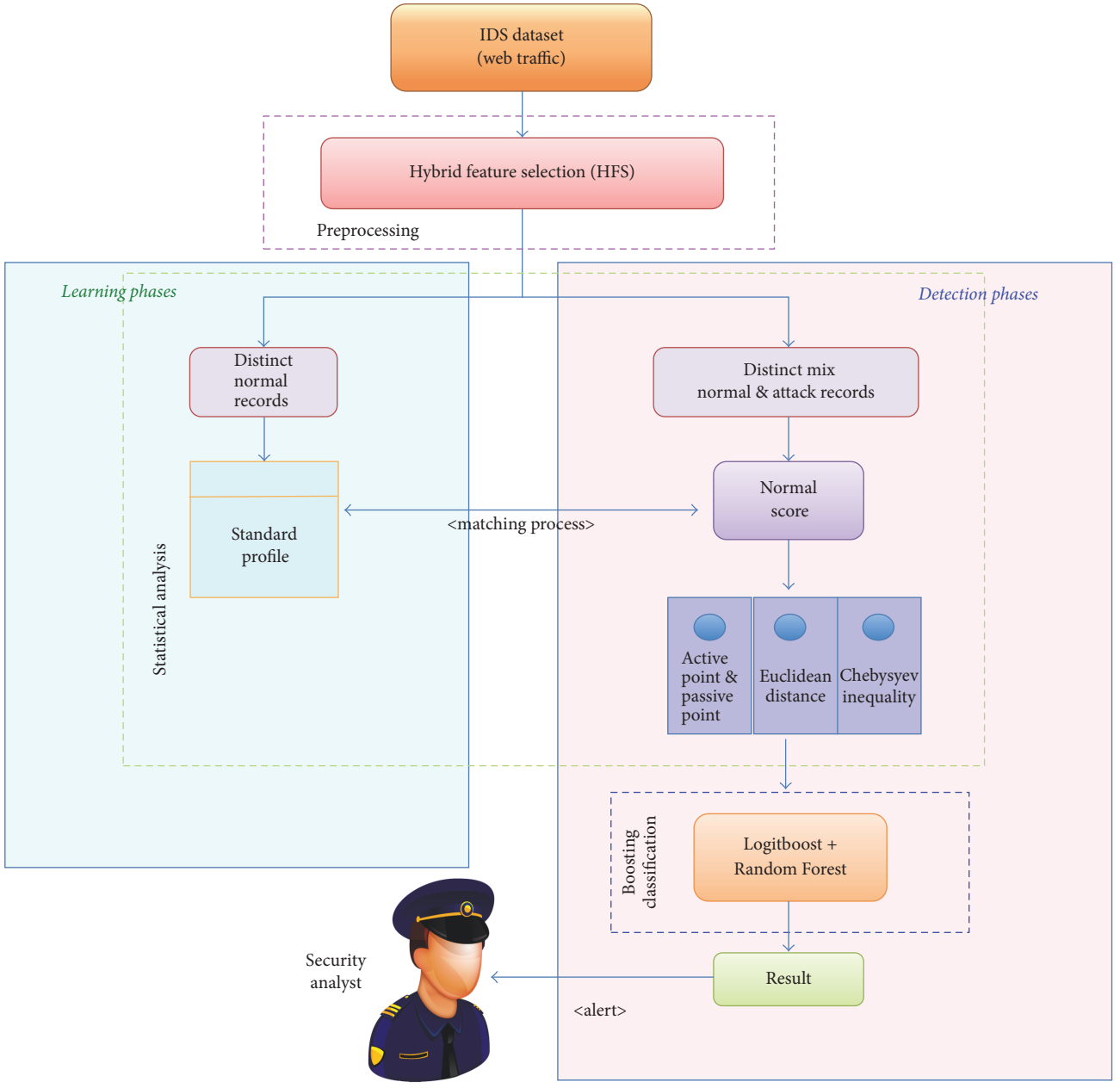

Figure 5: Proposed unified anomaly IDS.

the standard profile into the test dataset. All values within the test dataset were examined very closely. If their unique values are matched with the profile, a normal score will be awarded. However, if the test dataset values are absent in the standard profile, a zero score will be given to the particular attributes.

During the matching procedure, two scores, namely, the Passive Score (PS) and the Active Score (AS), are produced. PS is a fixed score obtained directly from the standard profile, while AS is generated during the matching procedure between the testing data and the standard profile.

Both scores collected during the detection phase (matching process) were then converted into data points that represent coordinates for distance measurement. Later, the degree of normality is defined by calculating the distance between the passive and active data points of the testing dataset. Figure 7 presents the example of normal traffic behaviour when both the passive point and the active point shared the same coordinates. In addition, Figure 8 presents the example of anomaly traffic behaviour when active points are separated from passive points and some outliers are produced. To measure the distance between these two points (benign and outliers), Euclidean Distance Analysis (EDA) was used, given its adequacy in computing basic distances. In this research, we make the assumption that anomalies will occur when there is deviation between normal and abnormal behaviour. Based on that assumption, we flag the possible

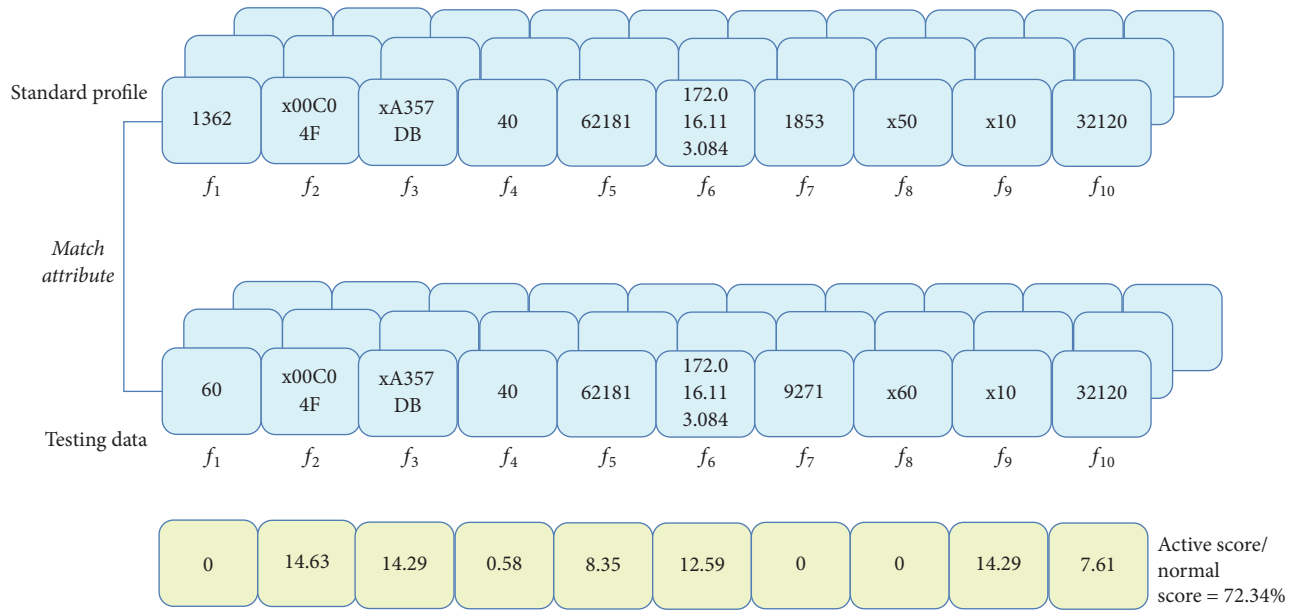

Figure 6: Matching attributes between standard profile and test data (DARPA 1999).

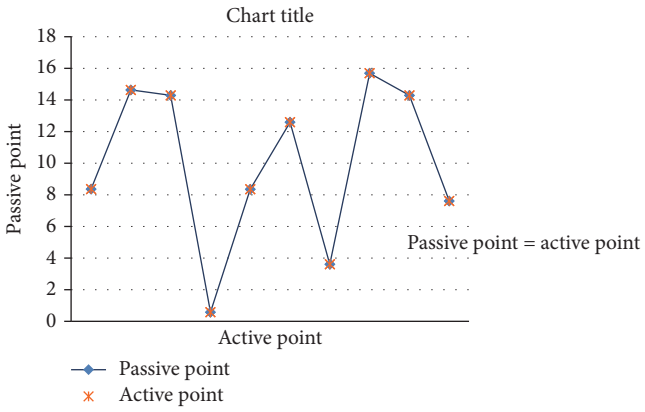

Figure 7: Normal packet behaviour (DARPA 1999).

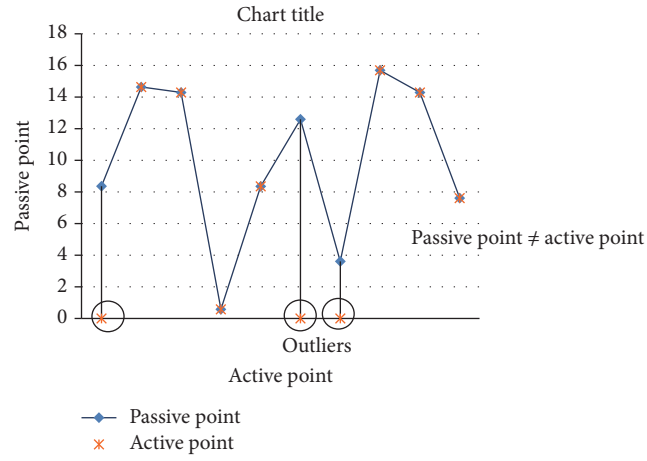

Figure 8: Anomaly packet behaviour (DARPA 1999).

intrusion by calculating the distance between the two points of training and testing data using Euclidean Distance. As we implement the rigid assumption, the false detection is expected to be huge. Thus further analysis using Chebyshev Inequality is deployed to measure the upper bound for threshold measurement that could improve the detection performance. The EDA between passive and active data points is computed as

$$\text{Euclidean Distance} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}. \quad (4)$$

Thus, the distance between passive point and active point can be simplified into

$$\text{Distance}_{ap} = \sqrt{\sum_{k=1}^{n} (x_a - x_p)^2}. \quad (5)$$

In the next process, we had considered packet size as an additional measure in conjunction with the standard profile. The justification of choosing this feature is briefly explained in the next subsection.

4.2.2. Influence of Packet Size on Traffic Behaviour. Previous work [54–56] has proven that the packet size (bytes) can be used to measure the traffic normality. This fact is validated by the nature of a client-server input service request. Typically, in client-server access, a client request would be comprised of a small number of bytes. In return, the server will respond with a large number of bytes. As such, a large number of requests can be considered or suspected as abnormal requests. Normally, when the user makes a request from the same source address, the increase of the extracted string packet size such as "get request" is minimal.

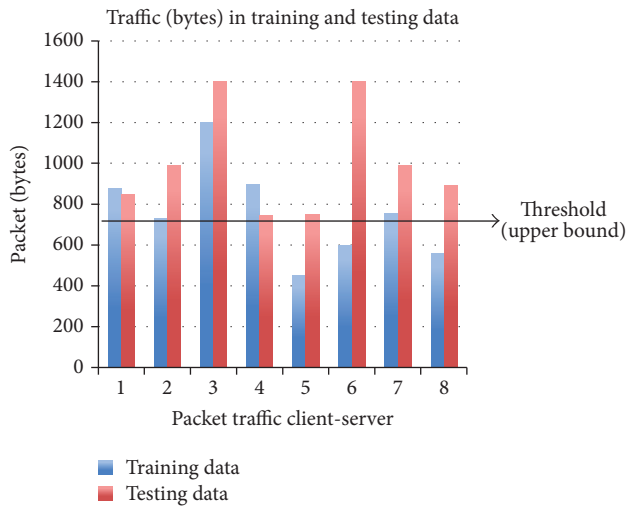

Figure 9: Example of traffic (bytes) flow from client to server.

For that reason, the inconsistent input size would trigger anomaly activity. This normally happens when malicious input is bound together in the legitimate traffic. For instance, the XSS (one of the top web attacks) would target web pages with an attempt to add malicious scripts to the website. This activity requires more data that significantly exceeded the size of the average parameter. With regard to the SQL injection type of attack, the attacker's input would include malicious code to misdirect the program execution. The code is in special strings that could alter the SQL statement with the intention to compromise the intended database files. Consequently, the malicious packets may contain up to several thousand bytes. We therefore statistically measure the packet size of the anomalous source traffic, which was first flagged as anomalous using EDA.

4.2.3. Threshold. We deployed the Chebyshev Inequality method to find the right boundary and determine the finest threshold to achieve a higher detection rate. We have considered anomalous source traffic and packet size as the main features in defining the threshold. Figure 9 illustrates an example of measuring the upper bound of training data from the mean using Chebyshev Inequality.

From the previous anomalous source IP address, we estimated the mean and the standard deviation of their packet size (bytes) distribution by determining the sample (mean and variance) of each parameter size $(S_1, S_2, S_3, \ldots, S_n)$ for both datasets during the learning phase (normal traffic). The mean and the variance collected during the learning phase were used to find the regularity in the detection phase. We measured the probability of a packet becoming irregular using Chebyshev Inequality as shown as follows:

$$P\left(\left|x-\mu\right|\right) \geq \tau \leq \frac{\sigma^2}{\tau^2}. \tag{6}$$

The advantage of using Chebyshev Inequality is that it does not rely on the knowledge of how the data is distributed, as in the real environment the traffic distribution could vary. It places an upper bound on the possibility that the deviation between the value of the random variables x and μ is greater than the threshold τ for random distribution with variance σ^2 and mean μ. We changed the threshold τ with the difference between the feature size S and the mean μ of the feature size distribution. This will define the upper bound for the probability that the feature size of a particular source IP address deviates more from the mean when compared with normal traffic. The probability value $P(S)$ for feature size S is calculated as follows:

$$P\left(\left|x-\mu\right|\right) \geq \left|S-\mu\right| \leq P\left(S\right) = \frac{\sigma^2}{\left(S-\mu\right)^2}. \tag{7}$$

4.3. Boosting Classification Algorithm (Second-Stage Detection). In the previous stage, the combination of EDA and CIT statistical approaches had demonstrated some attack detection performance (true positive) ability. However, when the detection approach is solely dependent on normal behaviour as a benchmark, massive false detection is produced. To reduce the false detection rate, the data mining approach is proposed. The main intention is to reexamine the traffic, which has been predicted in binary form either as an anomaly or normal. Furthermore, some additional features such as predicted_field, anomaly_field, and normal_fields which are generated in the first stage are induced into data mining techniques. The additional feature would improve the discriminative power of the classification algorithm, thus improving the detection capabilities and reducing the false alarm rate. In the data mining approach, we propose to use an ensemble technique named *boosting algorithm* that has the potential to improve the detection accuracy while minimising the false alarm rate, as it is proven to be more efficient than using a single algorithm [57].

In this research we use the boosting algorithm named *Logitboost* as the metaclassifier for boosting classification. From the literature, we found that this algorithm is more suitable in handling noisy and outlier data compared to the famous Adaboost algorithm. Consider a training data set with N samples and divided into two classes (in this study the two classes are abnormal and normal). The two classes are defined as $y \in \{-1, +1\}$; that is, samples in class $y = 1$ are normal traffic while $y = -1$ are the sample of attack traffic. Let the set of training data be $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$, where x_i is the feature vector, and y_n is the target class. The Logitboost algorithm consists of the following steps [39]:

(1) Input data set $N = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$, where $x_i \in X$ and $y_i \in Y = \{-1, +1\}$. Input number of iterations K.

(2) Initialise the weights $w_i = 1/N$, $i = 1, 2, \ldots, N$; start committee function $F(x) = 0$ and probabilities estimates $P(x_i) = 1/2$.

(3) Repeat for $k = 1, 2, \ldots, K$:

(a) Calculate the weights and working response

$$w_i = p\left(x_i\right)\left(1 - p\left(x_i\right)\right), \tag{8}$$

$$z_i = \frac{y_i - p(x_i)}{p(x_i)(1 - p(x_i))}. \quad (9)$$

(b) Fit the function $f_k(x)$ by a weighted least squares regression of z_i to x_i using weights w_i. In this research, we use Random Forest as weak classifier to fit the data using weights w_i.

(c) Update

$$F(x) \longleftarrow F(x) + \frac{1}{2} f_k(x),$$
$$p(x) \longleftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}. \quad (10)$$

(4) Output the classifier:

$$\text{sign}[F(x)] = \text{sign}\left[\sum_{k=1}^{K} f_k(x)\right]. \quad (11)$$

At this point, sign$[F(x)]$ is a function that has two possible output classes:

$$\text{sign}[F(x)] = \begin{cases} 1, & \text{if } F(x) < 0, \\ -1, & \text{if } F(x) \geq 0. \end{cases} \quad (12)$$

One of the key factors exerting influence on the performance of the boosting algorithm is the construction of the weak classifier. The weak classifier $f_k(x)$ chosen in (8) should be resistant to data overfitting and be able to manage data reweighing. Based on the successful performance of Random Forest (RF), we chose that algorithm as the weak classifier for Logitboost classification.

5. Experiment and Results

The detection performance of the proposed unified approach when applied to both the DARPA 1999 and the ISCX 2012 datasets is presented in this section.

The experimental results were obtained using the Waikato Environment for Knowledge Analysis (WEKA) data mining tools version 3.7 [58] and MySQL as a database management system. Three main performance metrics were used in this experiment to evaluate our proposed methods:

(a) *False Alarm Rate* (*FAR*). To quantify the amount of benign traffic detected as malicious traffic.

(b) *Detection Rate* (*DR*). The proportion of detected attacks among all attack data.

(c) *Accuracy* (*ACC*). Measured in percentage, where instances are correctly predicted

$$\text{False Alarm Rate (FAR)} = \frac{(\text{FP})}{(\text{FP}) + (\text{TN})},$$
$$\text{Detection Rate (DR)} = \frac{(\text{TP})}{(\text{TP}) + (\text{FN})}, \quad (13)$$
$$\text{Accuracy (ACC)} = \frac{(\text{TP}) + (\text{TN})}{(\text{TP}) + (\text{TN}) + (\text{FP}) + (\text{FN})}.$$

Table 3: DARPA 1999 dataset.

Dataset	Date	Normal traffic	Attack traffic
Training week 4	03/29/1999	8,998	728
	03/30/1999	101	643
	03/31/1999	5,202	456
	04/01/1999	11,413	605
	04/02/1999	0	0
	04/03/1999	0	0
	04/04/1999	0	0
Testing week 5	04/05/1999	6,632	723
	04/06/1999	6,873	993
	04/07/1999	5,800	1,807
	04/08/1999	77,039	640
	04/09/1999	0	8,073
	04/10/1999	174	62
Total		*136,962*	

Table 4: ISCX 2012 dataset.

Date	Training data		Testing data	
	Normal	Attack	Normal	Attack
6/11/2010	0	0	0	0
6/12/2010	528	0	2,074	0
6/13/2010	0	84	0	108
6/14/2010	826	873	782	1,096
6/15/2010	1,468	2,757	1,973	27,125
6/16/2010	432	0	1,237	0
6/17/2010	1,032	0	562	0
Total	*4,286*	*3,714*	*6,628*	*28,329*

We use the publicly available DARPA 1999 and ISCX 2012 datasets that represent traditional and modern intrusion datasets in evaluating our methods. The detail of the aforesaid datasets can be found in [30, 46]. In the DARPA 1999 dataset, the week 4 training data and week 5 testing data consist of 136,962 types of http traffic, as presented in Table 3. With regard to the ISCX 2012 dataset, 8,000 unique instances of http traffic were used in the training data while a total of 34,957 instances of http traffic were used in the testing data, as showed in Table 4.

In the preprocessing phase, we employed the HFS approach for both datasets to select the most prominent features. Through this process, the original 33 DARPA 1999 and the original 11 ISCX 2012 features were reduced to 10 and 4, respectively, as shown in Tables 5 and 6. This significant reduction of features has contributed to reducing the overall computational costs in this experiment.

Thereafter, the process continues to statistically measure the packet header with Euclidean Distance Analysis (EDA) and Chebyshev Inequalities. We used EDA to find the outliers in the testing data by calculating the distance between testing data and training data. In determining the finest threshold, the upper bound was computed using the CIT method. Table 7 shows the comparison results achieved in statistical analysis for both datasets.

Table 5: Feature selection for DARPA 1999 dataset.

Feature selection approach	Number of features	Feature selection
Original features	33	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33$
Reduced features (hybrid feature selection)	10	$f1, f4, f5, f9, f10, f16, f20, f24, f25, f26$

Table 6: Feature selection for ISCX 2012 dataset.

Feature selection approach	Number of features	Feature selection
Original features	11	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11$
Reduced features (hybrid feature selection)	4	$f2, f3, f9, f11$

Table 7: Comparison between SBAD and unified approach.

Dataset	Method	False alarm rate	Detection rate	Accuracy
DARPA 1999	Statistical based anomaly detection (SBAD)	5.10%	75.20%	92.67%
	Unified Intrusion Anomaly Detection (UIAD)	*0.13%*	*95.84%*	*99.41%*
ISCX 2012	Statistical based anomaly detection (SBAD)	3.50%	99.81%	99.18%
	Unified Intrusion Anomaly Detection (UIAD)	*0.08%*	*99.66%*	*99.71%*

By implementing SBAD alone, the approach was seen to generate a number of false alarm rates. Upon closer investigation, we found that the false detection was derived from masquerade traffic where benign traffic shared the same behaviour with malign traffic and vice versa. Thus, the data mining technique particularly using a boosting algorithm classification is employed as a compliment to the SBAD to reduce the inaccurate classification rate.

Table 7 presents a data performance comparison between SBAD and the proposed unified approach. The result shows that the proposed UIAD had outperformed SBAD in terms of FAR, DR, and ACC. This has indicated that the anomaly detection components in the second stage are a good complement for attack detection in first stage. The implementation of 2-stage detection has significantly reduces the false alarm rate from 5.1% and 3.5% to 0.13% and 0.08% for both datasets, respectively. Although the detection rate of SBAD in the ISCX 2012 dataset is slightly better by 0.15%, in terms of the overall accuracy produced, the performance of UIAD is slightly ahead by 0.53%, along with a more than 43 times reduction of false alarm rate compared to SBAD.

To test the robustness of our proposed unified approach, we ensured that the attack traffic in both training and testing data was significantly different. In simple terms, this mean that the sample attack traffic used in the training data is not itself part of the testing data. In addition, we made sure that the proportion of attack traffic in the training data was less than the attack traffic in the testing dataset. For example, in this research 2,432 and 3,714 amounts of attack traffic were used in the training data to build the classification model while 12,298 and 28,329 amounts of attack traffic are available for detection in the DARPA 1999 and ISCX 2012 testing sets, respectively.

Table 8 lists 6 types of attack available in both weeks 4 and 5 from the DARPA 1999 dataset. The 4 types of attack existed in week 4 (training dataset) were *back, ipsweep, perl,* and *phf*. Subsequently on week 5 (testing dataset), 5 types of attack, *back, ipsweep,* and *perl* plus two new attacks named *secret and tcpreset,* were identified. Our unified approach successfully recognised 95.84% of attack instances in the testing dataset. The attack types with the highest detection rate are U2R (100.00%) and DATA (100%), followed by DoS (75.71%), and the lowest is the PROBE (67.56%). Upon closer analysis, we noticed that the poor performance of PROBE was due to the nature of the attack itself, which shares similar characteristics with normal traffic behaviour. As the nature of PROBE attacks is to gather system information and to discover known vulnerabilities, the relevant kind of traffic seems to be legitimate and is mostly classified as normal by the system. With regard to the DoS attack type, the low detection percentage of "back" attack was caused by the lack of samples available in the training dataset. The sample was 52 times smaller than the attack in the testing dataset. It is worth mentioning that our proposed unified approach successfully identified 2 new attacks name "tcpreset" and "secret" that were only present in the testing dataset, which indicated that our proposed unified approach is capable of detecting unknown attacks.

In the ISCX 2012 dataset, the attack class is represented in binary form (0, 1) either as normal or attack traffic. Thus, the analysis on the specific attack type in the dataset is not possible. As shown in Table 9, with a limited number of attacks available in training dataset, the system successfully recognises almost all the attacks in the testing dataset, with a 99.66% detection rate.

Tables 10 and 11 show the performance of our proposed unified model in terms of FAR, DR, and ACC compared to the previous methods tested on the DARPA 1999 and ISCX 2012 datasets. It should be noted that the comparisons are for reference only due to many researchers having used different proportions of traffic types, sampling methods, and preprocessing techniques. Although our proposed approach

Table 8: Detection result derived by unified approach for DARPA 1999 testing dataset.

Attack category	Attack name	Attack traffic in training dataset	Attack traffic in testing dataset	Attack traffic detected by unified approach	% age of detected attack traffic
DoS	back	25	1,300	*983*	*75.71%*
	tcpreset	—	5	*5*	
PROBE	ipsweep	106	598	*404*	*67.56%*
U2R	perl	1,677	10,333	10,333	*100%*
R2L	phf	624	—	—	—
DATA (New)	secret	—	62	*62*	*100%*
Total	—	2,432	12,298	*11,787*	—

Table 9: Detection result derived by unified approach for ISCX 2012 testing set.

Attack traffic in training dataset	Attack traffic in testing dataset	Attack traffic detected by unified approach	% age of detected attack traffic
3,714	28,329	*28,234*	*99.66%*

had achieved better performance in most of the cases, it cannot be claimed that the proposed method outperformed others. Nevertheless, our proposed approach has shown some detection ability with a robust performance in detecting unknown attack traffic.

In addition, it should be noted that we evaluated the performance of the proposed approach with some eminent state-of-the-art data mining algorithms used in IDS. Tables 12 and 13 display a comparison of performance metrics between our proposed approach and seven other data mining algorithms previously used by researchers in IDSs, including Naïve Bayes [59], Support Vector Machine [60], Multilayer Perceptron [61], Decision Table [62], Decision Tree [63], Random Forest [64], and Adaboost [36].

To choose a better combination for the Logitboost classifier from a set of single classifiers in terms of accuracy, detection rates, and false alarm rates, five single classifiers are evaluated individually as illustrated in Figures 10 and 11. This is a crucial aspect of our research because the algorithm choices need to be further reclassified with ensemble approaches for better detection performance. In the DARPA 1999 dataset, among five classifiers the accuracy, detection rate, and false alarm rate shown by RF are comparable with others. Although MLP had shown slightly better performance compared to RF, the time taken to build a classification model by MLP is 84 times longer than RF. Meanwhile, in the ISCX 2012 dataset, RF outperformed every single other classifier by achieving 99.68% detection accuracy. Thus, in our unified detection approach, we had chosen RF to ensemble with the Logitboost classifier for both the DARPA 1999 and ISCX 2012 datasets.

To compare the performance of the Adaboost ensemble with RF and our unified approach, a further experiment is performed as presented in Tables 12 and 13. Due to the

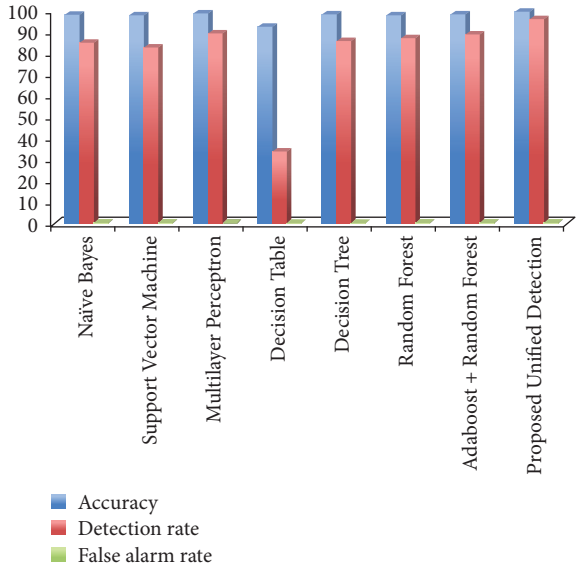

Figure 10: Comparison of performance algorithms in DARPA 1999 dataset.

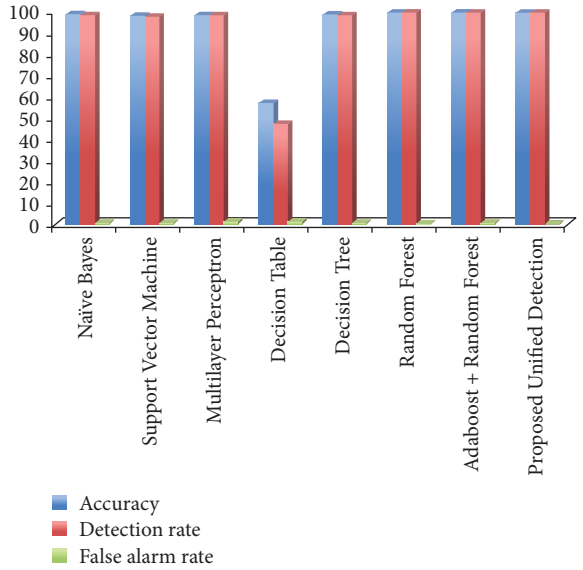

Figure 11: Comparison of performance algorithms in ISCX 2012 dataset.

Table 10: Comparison of FAR, DR, and ACC obtained by the proposed method and other previous works in DARPA 1999 dataset.

Methods	False alarm rate (%)	Detection rate (%)	Accuracy (%)
Improved IDS with fuzzy logic [10]	6.10	88.71	N/A
Lightweight IDS [6]	1.36	72.70	N/A
Ensemble neural classifier [11]	3.70	99.40	N/A
Sequential differentiate method [12]	3.38	100.00	N/A
Hybrid data mining [13]	2.75	97.25	N/A
Distribution IDS[14]	N/A	96.00	N/A
Catastrophe theory [15]	3.38	87.39	N/A
Unified Intrusion Anomaly Detection (2017)	*0.13*	*95.84*	*99.41*

Table 11: Comparison of FAR, DR, and ACC obtained by the proposed method and other previous works in ISCX 2012 dataset.

Methods	False alarm rate (%)	Detection rate (%)	Accuracy (%)
Packet Header Anomaly Detection [16]	N/A	99.04	N/A
SVM anomaly detection [17]	1.36	72.70	N/A
Computer vision techniques [18]	3.70	99.40	N/A
Payload based anomaly detection [19]	3.38	100.00	N/A
Evolved specialized ensembles [20]	N/A	91.37	N/A
Distributed SVM [21]	1.10	98.50	N/A
Unified Intrusion Anomaly Detection (2017)	*0.08*	*99.66*	*99.71*

high complexity of our proposed method, Table 12 indicated that our proposed method took slightly longer in building a classification model and attack detection compared to Adaboost + RF. As a result, our method took 0.82 seconds and 0.38 seconds longer than Adaboost + RF in building and testing classification model. Although our proposed method recorded higher computational complexity, overall performance that includes detection rate and overall accuracy rate reveals that our proposed method has indicated a better performance with 6.99% and 0.79% improvement, respectively, over the Adaboost + RF in DARPA 1999.

Table 13 presents the performance of our unified proposed approach on the ISCX 2012 dataset. Further comparison between Adaboost and our proposed unified approach has shown Adaboost performed slightly better in terms of performance accuracy and detection rate, displaying 27% less computational complexity in building classification model. However, the false alarm rates obtained by Adaboost are 6.5 times worse than our proposed model. From the aforementioned results, we conclude that our algorithm provides comparable detection accuracy rate with a low false alarm rate, which is the most crucial property of IDSs in practice.

6. Conclusion and Future Work

There were numerous anomaly intrusion detection studies made in the past. Nevertheless, achieving exceptionally low false alarm rates with high attack recognition capabilities for unseen attacks still remains a major challenge. This paper presented the novel Unified Intrusion Anomaly Detection (UIAD) experiment results. The experiment synthesised both statistical and data mining approaches to achieve better results. The model consists of three major parts: preprocessing, statistical measurements, and a boosting algorithm. The UIAD was evaluated using two publicly available labelled intrusion detection evaluation datasets (DARPA 1999 and ISCX 2012) to allow different integration testing environments. Initially, in the preprocessing phase, redundant and irrelevant features were filtered-out by HFS to obtain the most prominent features. Following that, we deployed the EDA and Chebyshev Inequality methods to measure and determine the normality (benign or malicious) of the traffic characteristics. We employed a data mining approach using the Logitboost classifier algorithm to improve the overall detection accuracy while reducing the false alarm rate. The combination detection of statistical analysis and data mining approaches demonstrated a promisingly reliable rate of anomaly based intrusion detection. Individually, the statistical approach was capable of demonstrating some level of detection ability. However, the better-synergised approach of statistical and data mining approaches yielded better performance particularly in reducing the low false alarm rate below 1%. The experimental results have demonstrated that our proposed UIAD has achieved comparable performance with other established state-of-the-art IDS algorithms. Moving forward, the final successful results will be transformed into signatures and stored in the blacklist database for future identification proposes. We believe that detection time can be drastically reduced, since the new entry traffic can be matched with benign/malicious signatures generated from the previous detection. Moreover, the proposed unified approach can potentially be evaluated online using larger, as well as the latest, encrypted sets of traffic.

Table 12: Comparison between proposed methods with other seven algorithms in DARPA 1999 dataset.

Algorithms	Model built (sec.)	Detection time (sec.)	False alarm rate (%)	Detection rate (%)	Accuracy (%)
Naïve Bayes (NB)	0.53	0.42	0.15	85.06	98.18
Support Vector Machine (SVM)	158	142	0.22	82.78	97.86
Multilayer Perceptron (MLP)	135	1.2	0.083	89.29	98.72
Decision Table (DT)	0.85	0.61	0.15	33.85	92.39
Decision Tree (J48)	0.97	0.67	0.05	85.84	98.35
Random Forest (RF)	1.6	1.13	0.17	87.10	98.39
Adaboost + Random Forest (RF)	3.41	1.83	0.13	88.85	98.62
Unified Intrusion Anomaly Detection (2017)	*4.23*	*2.21*	*0.13*	*95.84*	*99.41*

Table 13: Comparison between proposed methods with other seven algorithms in ISCX 2012 dataset.

Algorithms	Model built (sec.)	Detection time (sec.)	False alarm rate (%)	Detection rate (%)	Accuracy (%)
Naïve Bayes (NB)	0.06	0.22	0.26	98.94	99.09
Support Vector Machine (SVM)	35.22	34.41	0.35	97.92	98.25
Multilayer Perceptron (MLP)	22.83	0.5	1.16	98.71	98.74
Decision Table (DT)	0.17	0.11	1.15	47.75	57.44
Decision Tree (J48)	0.08	0.13	0.26	98.96	99.10
Random Forest (RF)	0.28	0.26	0.11	99.63	99.68
Adaboost + Random Forest (RF)	1.84	1.58	0.51	99.82	99.75
Unified Intrusion Anomaly Detection (2017)	*2.32*	*1.92*	*0.08*	*99.66*	*99.71*

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.

[2] S. V. Thakare and D. V. Gore, "Comparative study of CIA and revised-CIA algorithm," in *Proceedings of the 2014 4th International Conference on Communication Systems and Network Technologies, CSNT 2014*, pp. 713–718, April 2014.

[3] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-means clustering and OneR classification," in *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, pp. 192–197, December 2011.

[4] C.-M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," *Neurocomputing*, vol. 88, pp. 78–86, 2012.

[5] K. L. I. Iii, "Anomaly Detection for HTTP Intrusion Detection," in *Algorithm Comparisons and the Effect of Generalization on Accuracy*, p. 196, Anomaly Detection for HTTP Intrusion Detection, Algorithm Comparisons and the Effect of Generalization on Accuracy, 2007.

[6] C.-M. Chen, Y.-L. Chen, and H.-C. Lin, "An efficient network intrusion detection," *Computer Communications*, vol. 33, no. 4, pp. 477–484, 2010.

[7] P. Louvieris, N. Clewley, and X. Liu, "Effects-based feature identification for network intrusion detection," *Neurocomputing*, vol. 121, pp. 265–273, 2013.

[8] K. Leung, "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters," vol. 38, 2005.

[9] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[10] B. Shanmugam and N. B. Idris, "Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition, SoCPaR 2009*, pp. 212–217, December 2009.

[11] P. A. Raj Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Computer Communications*, vol. 34, no. 11, pp. 1328–1341, 2011.

[12] N. K. Raja, K. Arulanandam, and B. R. Rajeswari, "Two-level packet inspection using sequential differentiate method," in *Proceedings of the 2012 International Conference on Advances in Computing and Communications, ICACC 2012*, pp. 42–45, August 2012.

[13] B. Agarwal and N. Mittal, "Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques," *Procedia Technology*, vol. 6, pp. 996–1003, 2012.

[14] Z. Hakimi, K. Faez, and M. Barati, "An efficient architecture for distributed intrusion detection system," in *Proceedings of the 2013 10th International ISC Conference on Information Security and Cryptology, ISCISC 2013*, August 2013.

[15] W. Xiong, N. Xiong, L. T. Yang, J. H. Park, H. Hu, and Q. Wang, "An anomaly-based detection in ubiquitous network

using the equilibrium state of the catastrophe theory," *Journal of Supercomputing*, vol. 64, no. 2, pp. 274–294, 2013.

[16] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, Z. Muda, and H. Zulzalil, "Packet header anomaly detection using statistical analysis," *Advances in Intelligent Systems and Computing*, vol. 299, pp. 473–482, 2014.

[17] E. M. Nyakundi, *Using Support Vector Machines in Anomaly Intrusion Detection*, The University of Guelph, 2015.

[18] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection on denial-of-service attacks based on computer vision techniques," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 64, no. 9, pp. 2519–2533, 2015.

[19] M. Kakavand, N. Mustapha, A. Mustapha, and M. T. Abdullah, "Effective dimensionality reduction of payload-based anomaly detection in TMAD model for HTTP payload," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 8, pp. 3884–3910, 2016.

[20] G. Folino, F. S. Pisani, and P. Sabatino, "A distributed intrusion detection framework based on evolved specialized ensembles of classifiers," in *Applications of evolutionary computation. Part I*, vol. 9597 of *Lecture Notes in Computer Science*, pp. 315–331, Springer, [Cham], 2016.

[21] H. Huang, R. S. Khalid, and H. Yu, "Distributed Machine Learning on Smart-Gateway Network Towards Real-Time Indoor Data Analytics," in *Data Science and Big Data: An Environment of Computational Intelligence*, vol. 24 of *Studies in Big Data*, pp. 231–263, Springer International Publishing, Cham, 2017.

[22] S. Alelyani, J. Tang, and H. Liu, "Feature Selection for Clustering: A Review," *Data Clustering: Algorithms and Applications*, pp. 1–37, 2013.

[23] F. Amiri, M. M. R. Yousefi, and C. Lucas, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network & Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[24] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new hybrid filter–wrapper feature selection method for clustering based on ranking," *Neurocomputing*, vol. 214, pp. 866–880, 2016.

[25] M. A. Hall, *Correlation-based Feature Subset Selection for Machine Learning*, New Zeland, Hamilton, 1999.

[26] N. Cleetus and K. A. Dhanya, "Genetic algorithm with different feature selection method for intrusion detection," in *Proceedings of the 2014 1st International Conference on Computational Systems and Communications, ICCSC 2014*, pp. 220–225, December 2014.

[27] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[28] M. V. Mahoney and P. K. Chan, "PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic," Florida Technol. Tech. Rep. CS-2001, 2001.

[29] F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, "Modeling protocol based packet header anomaly detector for network and host intrusion detection systems," in *Proceedings of the 6th International Conference on Information Security*, Springer Berlin Heidelberg, 2007.

[30] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[31] U. Fiore, F. Palmieri, A. Castiglione, and A. de Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[32] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-Means clustering and Naïve Bayes classification," in *Proceedings of the 2011 7th International Conference on Information Technology in Asia: Emerging Convergences and Singularity of Forms, CITA'11*, July 2011.

[33] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 1–16, 2016.

[34] S. Fakhraei, H. Soltanian-Zadeh, and F. Fotouhi, "Bias and stability of single variable classifiers for feature ranking and selection," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6945–6958, 2014.

[35] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, part 2, pp. 119–139, 1997.

[36] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 2, pp. 577–583, 2008.

[37] M. Panda and M. R. Patra, "Ensembling rule based classifiers for detecting network intrusions," in *Proceedings of the ARTCom 2009 - International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 19–22, October 2009.

[38] W. Li and Q. Li, "Using naive Bayes with AdaBoost to enhance network anomaly intrusion detection," in *Proceedings of the 3rd International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2010*, pp. 486–489, November 2010.

[39] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[40] N. Laranjeiro, M. Vieira, and H. Madeira, "A learning-based approach to secure web services from SQL/XPath Injection attacks," in *Proceedings of the 16th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2010*, pp. 191–198, December 2010.

[41] T. Threepak and A. Watcharapupong, "Web attack detection using entropy-based analysis," in *Proceedings of the 2014 28th International Conference on Information Networking, ICOIN 2014*, pp. 244–247, February 2014.

[42] M. Zolotukhin, T. Hämäläinen, T. Kokkonen, and J. Siltanen, "Analysis of HTTP requests for anomaly detection of web attacks," in *Proceedings of the 12th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2014*, pp. 406–411, August 2014.

[43] M.-Y. Kim and D. H. Lee, "Data-mining based SQL injection attack detection using internal query trees," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5416–5430, 2014.

[44] A. Oza, K. Ross, R. M. Low, and M. Stamp, "HTTP attack detection using n-gram analysis," *Computers and Security*, vol. 45, pp. 242–254, 2014.

[45] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless Anomaly Detection Based on IEEE 802.11 Behavior Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2158–2170, 2015.

[46] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, pp. 357–374, 2012.

[47] P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman, "An overview of issues in testing intrusion detection systems," National Institute of Standards and Technology NIST IR 7007, 2003.

[48] G. Folino, F. S. Pisani, and P. Sabatino, "An incremental ensemble evolved by using genetic programming to efficiently detect drifts in cyber security datasets," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference, GECCO 2016 Companion*, pp. 1103–1110, July 2016.

[49] T. Kamarudin, M. H. Maple, and C. Watson, "Hybrid feature selection technique for intrusion detection system," *Int. J. High Perform. Comput. Netw*, 2016.

[50] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: a survey and results of new tests," *Pattern Recognition*, vol. 44, no. 2, pp. 330–349, 2011.

[51] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 38, no. 5, pp. 649–659, 2008.

[52] F. Attal, A. Boubezoul, L. Oukhellou, and S. Espie, "Powered two-wheeler riding pattern recognition using a machine-learning framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 475–487, 2015.

[53] T. M. Khoshgoftaar, C. Seiffert, J. Van Hulse, A. Napolitano, and A. Folleco, "Estimating Class Probabilities in Random Forest," in *Proceedings of the 6th International Conference on Machine Learning and Applications ICMLA*, pp. 348–353, 2007.

[54] J. M. Estévez-Tapiador, P. García-Teodoro, and J. E. Díaz-Verdejo, "Measuring normality in HTTP traffic for anomaly-based intrusion detection," *Computer Networks*, vol. 45, no. 2, pp. 175–193, 2004.

[55] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," *Computer Networks*, vol. 48, no. 5, pp. 717–738, 2005.

[56] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig, "Intrusion detection for encrypted web accesses," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, pp. 569–576, can, May 2007.

[57] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, no. 1, pp. 3–17, 2014.

[58] WEKA, "Weka 3: Data Mining Software in Java," http://www.cs.waikato.ac.nz/ml/weka.

[59] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1937–1946, 2014.

[60] S. Zaman and F. Karray, "Features selection for intrusion detection systems based on support vector machines," in *Proceedings of the 2009 6th IEEE Consumer Communications and Networking Conference, CCNC 2009*, January 2009.

[61] D. Parikh and T. Chen, "Data fusion and cost minimization for intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 381–389, 2008.

[62] P. Aditi and G. Hitesh, "A New Approach of Intrusion Detection System using Clustering, Classification and Decision Table," in *Proceedings of the International Conference on Advances in Computer Science and Application*, 2013.

[63] K. A. Jalil, M. H. Kamarudin, and M. N. Masrek, "Comparison of machine learning algorithms performance in detecting network intrusion," in *Proceedings of the 2010 International Conference on Networking and Information Technology, ICNIT 2010*, pp. 221–226, June 2010.

[64] A. Jain and L. Bhupendra, "Classifier selection models for intrusion detection system (IDS)," *Informatics Engineering, an International Journal (IEIJ)*, vol. 4, no. 1, pp. 1–11, 2016.