International Journal of Security and Its Applications
Vol. 9, No. 4 (2015), pp. 91-106
http://dx.doi.org/10.14257/ijsia.2015.9.4.10

A Novel Lightweight Hybrid Intrusion Detection Method Using a Combination of Data Mining Techniques

Jatuphum Juanchaiyaphum[1], Ngamnij Arch-int[2*], Somjit Arch-int[3] and Saiyan Saiyod[4]

[123]*Semantic Mining Information Integration Laboratory (SMIIL)*
[4]*Hardware-Human Interface and Communications Laboratory (H2I-Comm)*
Computer Science Department, Science Faculty, Khon Kaen University
Khon Kaen, 40002, Thailand
[1]*jatuphum@kkumail.com, {*[2*]*ngamnij,* [3]*somjit,* [4]*saiyan}@kku.ac.th*

Abstract

Hybrid intrusion detection systems that make use of data mining techniques, in order to improve effectiveness, have been actively pursued in the last decade. However, their complexity to build detection models has become very expensive when confronted with large-scale datasets, making them unviable for real-time retraining. In order to overcome the limitation of the conventional hybrid method, we propose a new lightweight hybrid intrusion detection method that consists of a combination of feature selection, clustering and classification. According to our hypothesis that there are different natures of attack events in each of network protocols, the proposed method examines each of network protocol data separately, but their processes are the same. First, the training dataset is divided into training subsets, depending on their type of network protocol. Next, each training subset is reduced dimensionally by eliminating the irrelevant and redundant features throughout the feature selection process; and then broken down into disjointed regions, depending on their similar feature values, by K-Means clustering. Lastly, the C4.5 decision tree is used to build multiple misuse detection models for suspicious regions, which deviate from the normal and anomaly regions. As a result, each detection model is built from high-quality data, which are less complex and consist of relevant data. For better understanding of the enhanced performance, the proposed method was evaluated through experiments using the NSL-KDD dataset. The experimental results indicate that the proposed method is better in terms of effectiveness (F-value: 0.9957, classification accuracy: 99.52%, false positive rate: 0.26%), and efficiency (the training and testing times of the proposed method are approximately 33% and 25%, respectively, of the time required for its comparison) than the conventional hybrid method using the same algorithm.

Keywords: *Hybrid intrusion detection; K-Means clustering; Decision tree; Feature selection*

1. Introduction

An intrusion detection system (IDS) plays a vital role in detecting various kinds of intrusions, defined as any set of actions that attempt to compromise the integrity, confidentiality, or availability; of resources in cyberspace [1]. Intrusion detection systems are categorized into two fundamental principles: *misuse*, and *anomaly* based detection [2, 3]. The approach of *misuse* detection is based upon the patterns of known attacks, and are often referred to as 'signature-based' attacks; as they generally rely on the rules for known attacks (or so-called 'signatures'). They are effective in detecting known attacks, and have a low false-alarm rate. However, they cannot detect novel attacks, in which

ISSN: 1738-9976 IJSIA

signatures do not exist. Conversely, detection approach through *anomaly* detection is based upon the detection of behavior (an attack) which deviates significantly from normal behavior. The techniques using for anomaly detection rely on the creation of knowledge profiles of normal behavior, in order to detect an attack. They therefore have the ability to detect unknown attacks, which would not be detected through misuse detection. However, if the profiles are too broadly defined, some attacks may escape detection; leading to a low detection rate. In contrast, if the profiles are too narrowly defined, some normal activities may be incorrectly defined as attacks. This raises false alarms [4].

In recent studies, many interesting approaches have been proposed by the use of data mining techniques to improve the quality of the IDS. Data mining techniques have proven to be advantageous in discovering knowledge useful in distinguishing intrusive behaviors from normal behaviors; through a process of searching for relationships and patterns within the network traffic data. Data mining has been applied to misuse-based detection, anomaly-based detection, and hybrid-based detection (which combines several methods together to improve the performance of the conventional IDS). Each of these approaches has achieved admirable success and proven effectiveness; which can be evaluated by detection rate, false alarm rate, and accuracy. However, when confronted with large-scale data, many hybrid-based approaches suffer from high computational burdens; especially those which are built using complex algorithms (e.g., ANN, SVM and SOM). Such algorithms typically involve expensive computations during the training process [5-7], and are also very time consuming in the selection of proper setting parameters [8, 9]. Moreover, there exists a detection overhead problem, owing to an increase in a number of detection processes in hybrid-based methods, which may lead to lower efficiency, generally measured by the response time during a network attack.

In the monitoring of network traffic, not every feature of data is relevant to classify the network intrusion. Therefore, many researchers, such as Li, Wang, Tian, Lu and Young [10], Sivatha Sindhu, Geetha and Kannan [11], Louvieris, Clewley and Liu [12], Guo, Zhou, Ping, Luo, Lai and Zhang [13], Zargari and Voorhis [14], and Amiri, Rezaei Yousefi, Lucas, Shakery and Yazdani [15]; have employed feature selection as a preprocessing phase to discover the optimal subset of features to be employed, instead of using all available features. The results have shown that feature selection can help to reduce computation complexity and improve the performance of IDS, due to the elimination of all irrelevant and redundant features.

We hypothesized that there are different nature of intrusion in each network protocol. If the network protocol data is examined separately, it will more effectively reduce noisy data, and improve prediction accuracy of the intrusion detection model. Owing to the different natures of attack events in each network protocol, the factors determining the intrusive activities should be different in each network protocol. If all of features which contain irrelevant and redundant features are examined, it may not only increase the time complexity of the classification model, but also deteriorate the performance of the classifiers. None of the above works proposed a scientific approach for separately discovering an optimal subset of each network protocol data.

In this paper, we propose a new lightweight hybrid intrusion detection method that combines data mining techniques, such as feature selection, clustering and classification. According to our hypothesis, the training data is divided into disjoint subsets, depending on their type of network protocol. Each subset is reduced dimensionality through the feature selection method, by removing irrelevant and redundant features, and then broken down into disjoint regions by the anomaly detection model. Finally, multiple misuse detection models are created for disjoint regions that deviate from the normal and anomaly regions, to refine the decision boundaries by learning the subgroups within the region.

The K-Means clustering is used to build the anomaly detection model, and multiple misuse detection models are created by the C4.5 decision tree. The CfsSubsetEval

attribute evaluator and the BestFirst search method are used to find the best feature set in the feature selection process. The proposed lightweight hybrid intrusion detection method was evaluated through experiments using the NSL-KDD dataset [16]. The experimental results indicate that the proposed method is better in terms of effectiveness and efficiency than the conventional hybrid method using the same algorithm.

The remainder of this paper is structured as follows: The second section provides a brief summary of related work. In the third section, the proposed method is introduced and explained in detail. The fourth section evaluates the performance of the proposed method, in terms of effectiveness and efficiency. The study concludes in the final section, with a summary and recommendation for future research.

2. Related Works

To overcome the limitation of traditional intrusion detection systems; various hybrid-based detection approaches, that combine machine learning techniques to improve the performance of the IDS, have been proposed and implemented.

Muniyandi, Rajeswari and Rajaram [17] developed "K-Means+C4.5"; a method devised to cascade K-Means clustering, as well as the C4.5 decision tree method; for classifying normal and anomalous activities. This cascading method is designed to alleviate the forced assignment and class dominance problem of the K-Means method, for classification in the anomaly detection system. The K-Means method first breaks down the training dataset into k subsets, using the Euclidean distance similarity. Next, multiple C4.5 decision tree models are created for the broken-down subsets. For each subset, the decision tree model refines the decision boundaries by defining the subgroups within each subset. Natesan, Balasubramanie and Gowrison [18] proposed an improvement of the single weak classifier, using AdaBoost (adaptive boosting machine learning algorithm). The classifiers such as Bayes Net, Naïve Bayes, and Decision Tree were used as weak classifiers. The results showed that the Naïve Bayes and Decision Tree classifiers performed better as a weak classifier, than those with AdaBoost. Nevertheless, the major drawback of these methods is that there is no mechanism for detection of novel attacks that do not have similar properties to the known attacks in the training dataset. This can cause a low detection rate when facing unknown test patterns that do not exist in the training dataset.

Depren, Topallar, Anarim and Ciliz [2] proposed the parallel hybrid method; utilizing both the anomaly and misuse detection module in tandem. The anomaly detection module uses a self-organizing map (SOM) to build the anomaly model, which detects the anomalous activity that deviates from normal behavior. The misuse detection module uses a C4.5 decision tree to classify various types of attacks. Each module is trained independently. After which, the decision-support system combines the classification results of both modules. Govindarajan and Chandrasekaran [7] presented the hybrid architecture involving ensemble and base classifiers for the intrusion detection system. Multilayer perceptron (MLP) and radial basis function (RBF) neural networks were used to build the classifier models. The ensemble module combines the classification results of both models, and makes final output predictions by considering the predicted probabilities of each model. The experiment results demonstrated that the performance of this method was superior to that of the single usage of an existing classification method, such as MLP or RBF. However, as a drawback of parallel hybrid methods, every observed connection is examined by each of the classifier models; which can raise the detection overhead.

Peddabachigari, Abraham, Grosan and Thomas [19] proposed the ensemble approach; which combines the individual base classifiers: namely, the decision tree (DT), the support vector machine (SVM), and the hierarchical hybrid model (DT-SVM), with the model of the intrusion detection system. In the hybrid DT-SVM model, the dataset is passed through the DT, which generates the node information. The training and testing

data, along with the node information, is given to the SVM. The SVM gives the final output of the hybrid DT-SVM. The final output among the base classifier outputs (DT, SVM, DT-SVM) is determined by the highest scoring class. Kim, Lee and Kim [20] proposed a hierarchical hybrid intrusion detection method, that integrates a misuse detection model and an anomaly detection model in a separate structure. Within this method, a misuse detection model is built, based upon the C4.5 decision tree. The normal training data is broken down into smaller subsets, and then multiple 1-class SVM models are built for each subset. As the training dataset is broken down into smaller subsets, the data patterns of each subset are less complex than those of the dataset as a whole. Multiple models for each of the smaller data patterns therefore, may be less flexible than a single model for the entire data pattern. Moreover, the training and testing times are significantly reduced. The parameter settings of SVM in these works are performed by expert users, since the quality of the SVM models depends on having the proper parameters (i.e., Kernel function, C, Gramma, etc.). Although many approaches have been proposed in order to reduce the time required to find a proper parameter of SVM [8, 9, 21], they are still computationally expensive. Moreover, they are unfavorable for large-scale datasets, as training complexity is greatly dependent on the amount of data in the training set [5]; which may lead to a delay in automatically retraining, on the fly.

In recent literature, many researchers have concentrated on combining several learning techniques in order to reach the highest detection rate, with a low false-positive rate of IDS. However, there is a detection overhead problem, owing to the increase in computational complexity. In order to overcome the limitation in the computational complexity of the hybrid-based detection method, we have incorporated K-Means clustering and feature selection technique to preprocess the data, prior to being processed by the detection classifier. This not only decreases the dimensionality of the data, but also avoids the over-fitting that occurs when the algorithm model picks up data with uncommon characteristics. The method we propose is faster and more effective, in the classification of known and unknown patterns.

3. Proposed Lightweight Hybrid Intrusion Detection Method

In this section, the lightweight hybrid intrusion detection (LHID) method, which uses a combination of data mining techniques, is proposed. As shown in Figure 1, the proposed method is divided into two phases: namely, a training phase and a testing phase. In the training phase, anomaly and misuse detection models are built from the training dataset, and used to examine the test instances in the testing phase. Each phase is composed of the preprocessing module, the anomaly detection module, and the misuse detection module. Each module is described in detail as follows.

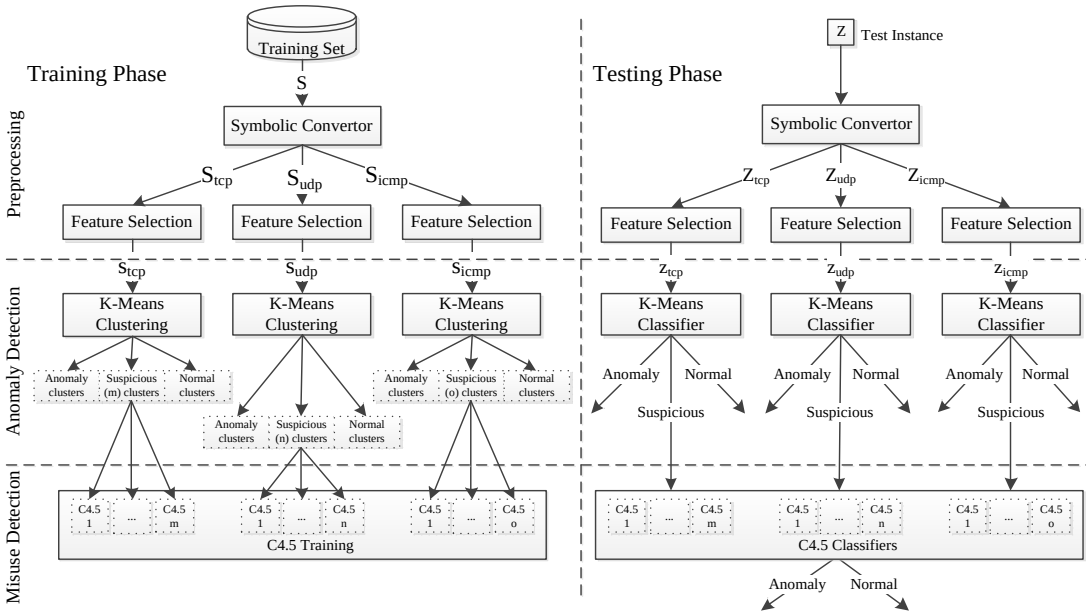

Figure 1. Architecture of Lightweight Hybrid Intrusion Detection Method

3.1. PreprocessingModule

The purpose of the preprocessing module is to prepare the data prior to being processed by the anomaly and misuse detection modules. This module is composed of two sub modules: the symbolic convertor, and the feature selection.

3.1.1. Symbolic Convertor

The symbolic convertor is a process which replaces the symbolic data of the dataset with numeric data, and divides the converted dataset into smaller subsets; according to their type of network protocol. For this research, we used the NSL-KDD standard dataset, consisting of 41 features, including both numeric (continuous) and symbolic (discrete) data; such as service type (e.g., HTTP, SMTP, FTP, etc.) and connection status flag (e.g., OTH, RSTO, REJ, etc.). The dataset must then be converted into numeric data, suitable for use with K-Means, for further processing. The symbolic convertor replaces the symbolic data with the proposed risk values [22]. The risk value is the numeric value, which reflects the risk of intrusion of the symbolic data. The risk values of service types and connection status flags are shown in Table 1 and Table 2.

After all symbolic data within the data set have been replaced the symbolic convertor divides the converted dataset into subsets, depending on their type of network protocol. Because there are different characteristics of attack events in each network protocol, if examined separately, it will reduce noisy data and enhance the performance of the IDS.

Table 1. Risk Value of Service Type

service	risk	service	risk	service	risk
Aol	1	echo	1	hostnames	1
Auth	0.31	eco_i	0.8	http	0.01
Bgp	1	ecr_i	0.87	http_2784	1
Courier	1	efs	1	http_443	1
csnet_ns	1	exec	1	http_8001	1
Ctf	1	finger	0.27	imap4	1
Daytime	1	ftp	0.26	IRC	0
Discard	1	ftp_data	0.06	iso_tsap	1
Domain	0.96	gopher	1	klogin	1
domain_u	0	harvest	1	kshell	1
Ldap	1	nntp	1	rje	1
Link	1	ntp_u	0	shell	1
Login	1	other	0.12	smtp	0.01
Mtp	1	pm_dump	1	sql_net	1
Name	1	pop_2	1	ssh	1
netbios_dgm	1	pop_3	0.53	sunrpc	1
netbios_ns	1	printer	1	supdup	1
netbios_ssn	1	private	0.97	systat	1
Netstat	1	red_i	0	telnet	0.48
Nnsp	1	remote_job	1	tftp_u	0
Time	0.88	uucp	1	X11	0.04
tim_i	0.33	uucp_path	1	Z39_50	1
urh_i	0	vmnet	1	urp_i	0
Whois	1				

Table 2. Risk Value of Connection Status Flag

flag	risk	flag	risk	flag	risk
OTH	0.729	RSTR	0.882	S3	0.08
REJ	0.519	S0	0.998	SF	0.016
RSTO	0.886	S1	0.008	SH	0.993
RSTOS0	1	S2	0.05		

3.1.2. Feature Selection

In a large stream of network traffic data, not every feature of the data is relevant to classify the intrusion. In order to make the IDS more efficient, feature selection is used to discover an optimal subset of features, rather than using all available features. This is achieved by combining a feature subset evaluator with a search method. The search method finds the best feature set, and the evaluator method then evaluates the worth of each subset of features. Throughout this process, the irrelevant and redundant features in each network protocol subset are eliminated. As a result, each subset is less complex and

less noisy. This not only reduces the computational times, but also enhances the classification performance of the detection models.

In this process, the BestFirst [23] is used to search the space of feature subsets by greedy hill-climbing augmented with a backtracking facility, and then the CfsSubsetEval [24] is used to evaluate the worth of a subset of features, by considering the individual predictive ability of each feature, along with the degree of redundancy between them; subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

Throughout the feature selection process, the number of dimensionality of each subset is decreased. In this research, the feature set for each network protocol subset is selected from the 41 available features, namely: 11 features for the TCP subset; 7 features for the UDP subset; and 4 features for the ICMP subset. The detail of selected features of each network protocol subset is presented in Table 3.

Table 3. The Features Selected by BestFirst+CfsSubsetEval

Dataset	Selected features
TCP subset	duration, service, src_bytes, dst_bytes, num_failed_logins, rerror_rate, srv_count, srv_serror_rate, srv_rerror_rate, st_host_same_src_port_rate, dst_host_srv_rerror_rate
UDP subset	duration, src_bytes, dst_bytes, land, dst_host_diff_host_rate, dst_host_srv_serror_rate, dst_host_srv_rerror_rate
ICMP subset	src_bytes, count, srv_serror_rate, dst_host_srv_rerror_rate

3.2. Anomaly Detection Module

The purpose of the anomaly detection module is to distinguish normal from anomalous behavior, as well as to reduce the possibility of over-fitting; which happens when algorithm models pickup data with uncommon characteristics. This is achieved through K-Means [25], a similarity based clustering, which divides training data into smaller regions according to their feature values. Throughout the module, each region is less complex, and consists of common characteristics with similar feature values. Subgroups or overlaps occurring within a region are not classified as normal or anomaly, and are labeled as suspicious regions. The regions then train the models, by misuse detection module. The anomaly detection module is described as follows.

The K-Means clustering partitions n data points on their feature values into k disjoint clusters. Where k is a positive integer number specifying the number of clusters, and has to be given in advance. The steps in anomaly detection using K-Means clustering are as follows:

1. Define the number of clusters k and arbitrarily choose an initial k cluster centroid
   $C = \{c_1, c_2, ..., c_k\}$.
2. For each training instance X:
   a. Compute the Euclidean distance
      $D(c_i, X), i = 1...k$.
      Find cluster c_q that is closest to X.
   b. Assign X to c_q. Update the centroid of c_q. (The centroid of a cluster is the arithmetic mean of the instances in the cluster.)
3. Repeat Step 2 until C does not change any more.
4. For each testing instances Z:

a. Compute the Euclidean distance
$D(c_i, Z), i = 1...k$.
Find cluster c_r that is closest to Z.

b. Classify Z as a normal, an anomaly, or a suspicious instance using the threshold rule. The threshold rule for classifying a testing instance Z that belongs to cluster c_r is:

$$Assign\ Z \rightarrow \begin{cases} 0, if\ \ P(\omega_{1r} \mid \mathrm{Z} \in \mathrm{c_r}) - P(\omega_{0r} \mid \mathrm{Z} \in \mathrm{c_r}) = -t \\ 1, if\ \ P(\omega_{1r} \mid \mathrm{Z} \in \mathrm{c_r}) - P(\omega_{0r} \mid \mathrm{Z} \in \mathrm{c_r}) = t \\ 2, otherwise \end{cases} \quad (1)$$

Where 0, 1, and 2 represent normal, anomaly, and suspicious classes; ω_{0r} and ω_{1r} represent the normal and anomaly classes in cluster $\mathrm{c_r}$; $P(\omega_{0r} \mid \mathrm{Z} \in \mathrm{c_r})$ and $P(\omega_{1r} \mid \mathrm{Z} \in \mathrm{c_r})$ represent the probability of normal and anomaly instances in cluster $\mathrm{c_r}$; and t is a predefined threshold. In this research, the threshold is set to 1.0.

In the training phase, before building the anomaly detection model, the parameter (k) of the K-Means clustering needs to be optimized. The data in the network protocol subsets is used to find the optimal value of the parameter k. Then, the value is selected to build the anomaly detection model for each of the network protocol subsets (e.g., 14 for TCP, 3 for UDP and 3 for ICMP). Each anomaly detection model divides its data into smaller regions, according to their similar feature values, and labels the regions as the classes of condition belonging to each region (equation 1). Throughout this phase, each region represents the label as normal, anomaly, or suspicious. The regions are then used to examine the test instances in the testing phase.

In the testing phase, the anomaly detection model classifies the test instance as the label of region to which its feature value is closest. Test instances determined to be suspicious are re-examined in the misuse detection module.

3.3. Misuse Detection Module

In order to refine the label of the suspicious instances, which cannot be classified by the anomaly detection module; the misuse detection module builds multiple misuse models from suspicious regions, obtained from the anomaly detection module. Throughout the process, the misuse detection model uses its decision function to label each suspicious instance, as normal or as an anomaly.

The C4.5 decision tree [26] is one of the most widely used, and practical methods for inductive inference. It is an enhancement of the ID3 algorithm [27] that has additional features, such as handling missing values, categorization of continuous attributes, pruning of decision trees, rule derivation, and so on. The aim of C4.5 is to recursively partition data into sub-groups, and building decision trees, in a top-down recursive divide-and-conquer manner. A tree is constructed by finding the highest information gain attribute test to conduct, at the root node of the tree. After the test is chosen, the cases are split according to the test, and the sub-problems are solved recursively. The attribute with the highest information gain is calculated using formulas 2 and 3.

$$Entropy(S) = -\sum_{i=1}^{n} Pr(C_i) log_2 Pr(C_i) \quad (2)$$

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{m} Pr(A_i) Entropy(S_{A_i}) \quad (3)$$

Where $Entropy(S)$ is information entropy of set S; n is the number of different classes in set S; $Pr(C_i)$ is the frequency of class C_i in S; $Gain(S,A)$ is the gain of set S after a split on A attribute; m is the number of difference values of attribute A in S; $Pr(A_i)$ is the frequency of classes that have A_i value in S; and $Entropy(S_{A_i})$ is a subset of S containing all items that have A_i value.

In the training phase, the multiple misuse detection models provided by the C4.5 are built for the suspicious regions and refine the decision boundaries, by identifying the subgroups within the region. Because the data patterns of the suspicious regions are less complex, and their similarity among data members within the same region are higher than those of the entire dataset; multiple classifiers built from suspicious regions can firm the classification task and correctly classify the observed activities into response class, much ahead of a single classifier. Moreover, the great length of time and complexity of the training process can also be reduced as the suspicious regions have less dimensionality than those of the whole dataset.

In the testing phase, the test instances determined as suspicious activity are classified by the appropriate classifier based on the region in which they belong. The instances that match normal patterns are determined as normal activities. Others are determined as anomalous activities.

4. Experimental Results

In this section, we demonstrated the performance of the proposed method. The NSL-KDD dataset was used as the benchmark dataset. To be easily and fairly compared with one another, all machine learning techniques examined in this paper were implemented in Java programming language, and WEKA 3.7 API [28]. All experiments were conducted on a machine with an Intel Core i7, 3.40 GHz, and 8 GB of RAM; running on Windows 7, SP1.

4.1. Dataset

The performance of the proposed method was evaluated through experiments using the NSL-KDD [16] dataset, which is an improved version of Knowledge Discovery and Data Mining (1999) dataset (KDD'99) [29]. The analysis of the KDD'99 dataset [16] found that there was an inherent problem of a number of redundant instances existing in the training and testing datasets, which greatly affected performance, resulting in poor evaluation of the anomaly detection methods. To solve this problem, the NSL-KDD dataset was proposed, removing all redundant instances, and reconstructing the dataset; which provided a more accurate and efficient evaluation of the various learning techniques. In this research, the evaluation dataset was organized by modifying the NSL-KDD as follows:

4.1.1. Training Data

The training dataset is organized by the KDDTrain+.TXT document in the NSL-KDD dataset. The full NSL-KDD training set consists of 125,973 instances.

4.1.2. Testing Data

The unknown attack data set was organized by modifying the KDDTest+.TXT. Because the KDDTest+.TXT contained instances similar to those existing in

KDDTrain+.TXT, the instances were removed. The evaluation data set is described in detail, in Table 4.

Table 4. Characteristics of the Evaluation Dataset

Training Data 125973 instances			Testing Data 21927 instances		
Protocol type	Class		Protocol type	Class	
	Normal	Anomaly		Normal	Anomaly
TCP	53600	49089	TCP	7833	10832
UDP	12434	2559	UDP	1706	842
ICMP	1309	6982	ICMP	85	629

4.2. Performance Metrics

To evaluate the performance of the proposed method, six widely used performance metrics are applied. They are the detection rate (DR, also known as the true positive rate or sensitivity); precision (PR); F-value; false positive rate (FPR, also known as a false alarm rate); accuracy (ACC); and Area Under an ROC Curve (AUC). The performance metrics was applied and calculated, using the confusion matrix given in Table 5

Table 5. Confusion Matrix

Actual Class	Predicted Class	
	Negative Class (Normal)	Positive Class (Anomaly)
Negative Class (Normal)	True negative (TN)	False positive (FP)
Positive Class (Anomaly)	False negative (FN)	True positive (TP)

These matrices are defined in the following equations:

$$DR = \frac{TP}{TP + FN} \tag{4}$$

$$PR = \frac{TP}{TP + FP} \tag{5}$$

$$F - score = \frac{(1+\beta^2) * DR * PR}{(\beta^2 * PR) + DR} \quad where\ \beta = 1 \tag{6}$$

$$FPR = \frac{FP}{TN + FP} \tag{7}$$

$$ACC = \frac{TN + TP}{TN + TP + FN + FP} \tag{8}$$

4.3. Results and Discussion

In this research, there are two key factors of evaluation: namely, effectiveness and efficiency. The effectiveness can be evaluated by F-value, DR, PR, AC, FPR and AUC, while the efficiency is measured by the average execution time in the training and testing process. To better understand the enhanced performance, the proposed method (LHID) was compared with both the conventional methods (i.e., C4.5 and K-Means+C4.5) and

lightweight conventional methods (i.e., lightweight C4.5 and lightweight K-means+C4.5), which are the combinations of the conventional methods and a feature selection method. The experimental results are as follows:

4.3.1. Effectiveness

The results presented in Table 6 compare the overall (detection and classification) performance of the proposed method to that of the two conventional methods. Overall, the results indicate that the proposed method achieves better performance in terms of F-value, DR, PR, AC, FPR and AUC (as Figure 2 shows the ROC curves of the proposed method and its comparisons).

Table 6. A Comparison between Results of the Proposed LHID Method, C4.5 and K-Means+C4.5 for Unknown Attack Data

Methods	Features	F-value	DR	PR	ACC	FPR	AUC
C4.5	41	0.7843	0.6624	0.9613	0.7956	0.0188	0.8141
Lightweight C4.5	11	0.8330	0.7309	0.9683	0.8356	0.0160	0.8502
K-Means+C4.5	41	0.8379	0.7362	0.9720	0.8401	0.0142	0.8546
Lightweight K-Means+C4.5	11	0.9162	0.8546	0.9874	0.9123	0.0067	0.9203
LHID	11,7,4	**0.9957**	**0.9959**	**0.9955**	**0.9952**	**0.0026**	**0.9951**

In order to demonstrate the effectiveness of the feature selection method, we compared the conventional methods with the lightweight conventional methods. In terms of the classification accuracy (ACC), the results show that the lightweight C4.5 is 4% better than the C4.5, and the lightweight K-Means+C4.5 is 7.22% better than the K-Means+C4.5. Moreover, the lightweight C4.5 is 0.28% lower than the C4.5, and the lightweight K-Means+C4.5 is 0.75% lower than the K-Means+C4.5, in terms of the false positive rate (FPR).

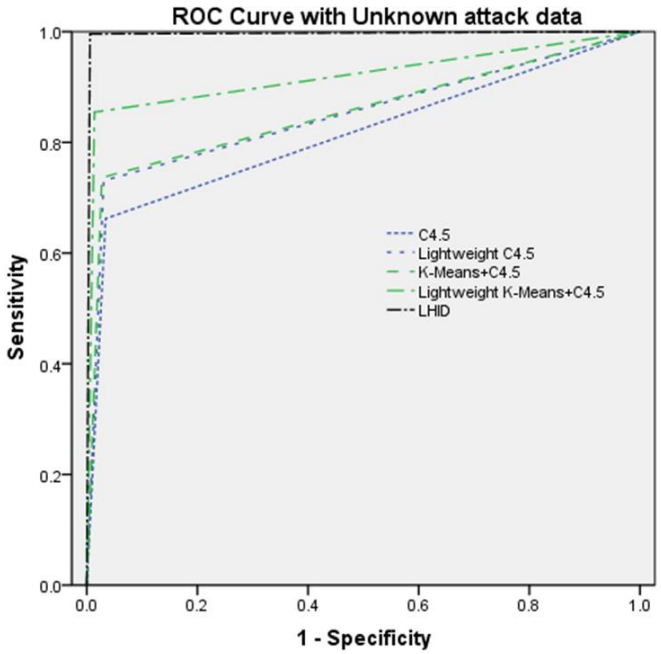

Figure 2. ROC Curves C4.5, K-Means+C4.5 and LHID Method over Unknown Attack Data

To demonstrate the effectiveness of the proposed method, we compared the proposed method with the conventional methods, the results indicate that the proposed method is 19.96% better than the C4.5, 15.96% better than the lightweight C4.5, 15.51% better than the K-Means+C4.5, and 8.29% better than the lightweight K-Means+C4.5, in terms of ACC. In terms of F-value, the proposed method is 21.14% better than the C4.5, 16.27% better than the lightweight C4.5, 15.78% better than the K-Means+C4.5, and 7.95% better than the lightweight K-Means+C4.5. In terms of FPR, the proposed method is 1.62% lower than the C4.5, 1.34% lower than the lightweight C4.5, 1.16% lower than the K-Means+C4.5, and 0.41% lower than the lightweight K-Means+C4.5.

The proposed method uses the same algorithm as the lightweight K-Means+C4.5. However, the method is different. The lightweight K-Means+C4.5 uses a whole dataset to build the intrusion detection models, whereas the proposed method uses broken-down subsets, which are grouped by their type of network protocols, to build the intrusion detection models. Thus, the results confirm our hypothesis that breaking down data into smaller subsets, according to their type of network protocols, can help to improve the classification detection performance of the IDS.

4.3.2. Efficiency

The results on the computational time in training and testing of each method are shown in Table 7, averaged over 30 runs.

Table 7. A Comparison between Computational Times of the Proposed LHID, C4.5 and K-Means+C4.5

Methods	Training time (s)	Testing time (s)
C4.5	24.22±1.488	0.121±0.003
Lightweight C4.5	**4.9±0.288**	**0.036±0.003**
K-Means+C4.5	41.27±0.211	0.247±0.014
Lightweight K-Means+C4.5	13.62±0.575	0.063±0.005
LHID	13.67±0.146	0.061±0.003

In order to illustrate the efficiency of the feature selection method, the conventional methods are compared with the lightweight conventional methods. The results indicate that the training and testing times of the lightweight C4.5 are approximately 20% and 30% of the time required for the C4.5, respectively, while the training and testing times of the lightweight K-Means+C4.5 are only approximately 33% and 26% of the time required for the K-Means+C4.5, respectively.

In order to illustrate the efficiency of the proposed method, we compared the proposed method with the conventional hybrid method (K-Means+C4.5), which uses the same algorithm. The results indicate that the training and testing times of the proposed method are only approximately 33% and 25% of the time required for the conventional hybrid method, respectively, using the same algorithm.

Figure 3 and Figure 4 show the comparison of the computational time in training, and the testing of all methods, respectively. The results show that the error bars of the proposed method and the lightweight K-Means+C4.5 method overlap. In order to measure the diversity of the computational time between these two methods, the independent samples *t*-test was used, to assess whether the means of these two methods are statistically different from each other. With a confident level of 99% ($\alpha = 0.01$), we obtain $p = 0.647$ ($t = -0.462$, df = 32.747) for training time, and $p = 0.029$ ($t = 2.244$, df = 58) for testing time. According to the *t*-test, the difference of computational time between the proposed method and the lightweight K-Means+C4.5 method in both cases is not statistically significant ($p > 0.01$). It should be noted that the proposed method does not require an

additional overhead in order to integrate the detection models, although it is more complex than its comparison.

Since the proposed method divides the training dataset into subsets, according to their type of network protocols, throughout the feature selection process, the dimensionality (number of instances × number of features) of the training dataset is 1,129,579 for the TCP subset; 104,951 for the UDP subset; 33,164 for the ICMP subset; and 1,267,694 for the total dataset, while the number of dimensionality of the training data of the lightweight K-Means+C4.5 is 1,385,703. Thus, the dimensionality of the proposed method is smaller than the lightweight K-Means+C4.5. Moreover, in the misuse detection module, the proposed method builds the misuse models for suspicious regions, and examines some instances that are determined as suspicious activity, whereas the lightweight K-Means+C4.5 builds misuse models for all subsets, and examines all instances. This is the reason why the computational time of the proposed method is still no different from the lightweight K-Means+C4.5, which is a combination of the K-Means+C4.5 and the feature selection method.

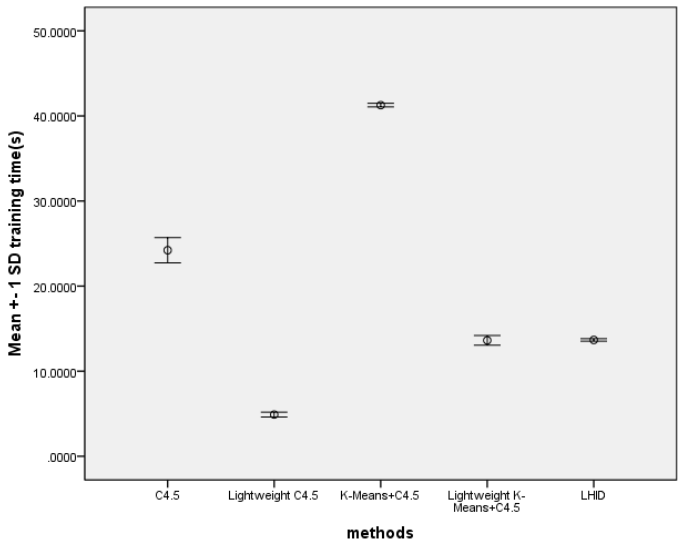

Figure 3. Computational Times when Training on NSL-KDD Dataset

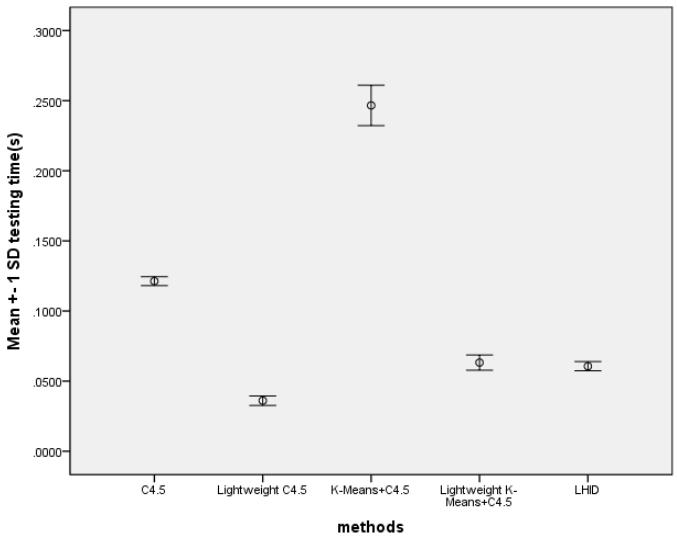

Figure 4. Computational Times when Testing on NSL-KDD Dataset

5. Conclusion

In this paper, we have proposed a new lightweight hybrid intrusion detection method that combines data mining techniques, such as feature selection, clustering and classification. Owing to our hypothesis that there are different natures of attack events in each network protocol, the proposed method examined each network protocol's data separately, but their processes were the same. The training dataset was divided into smaller subset, according to their type of network protocol. Each training subset was reduced the dimensionality by eliminating the irrelevant and redundant features throughout the feature selection process; and then divided into disjoint regions, according to their similar feature values, by the anomaly detection model, based on the K-Means clustering. Lastly, the C4.5 decision tree was used to build multiple misuse detection models for suspicious regions, which deviate from the normal and anomaly regions to refine the decision boundaries, by identifying the subgroups within each region.

The experimental results indicated that the proposed lightweight hybrid intrusion detection method can not only detect network attack efficiency, but also provide comparable detection performance in terms of F-value, DR, PR, AC, FPR and AUC, when compared with the conventional hybrid method using the same algorithm. The results also confirmed our hypothesis that examining each of network protocol's data separately can help to improve the performance of the IDS.

The proposed lightweight hybrid intrusion detection method achieved greater effectiveness and efficiency through the provision of high-quality training subsets, which are less complex than the entire training dataset; and maintained higher similar feature values within the subset. This not only decreased the dimensionality of the data, but also avoided the over-fitting that may occur when the algorithm model picks up data with uncommon characteristics. Our method therefore, tested faster and more effective in classifications of both known and unknown patterns.

Although the proposed method demonstrated admirable performance in the detection of network intrusions, it is designed primarily for detecting simple attacks. It does not detect the complex attacks, which in structure, are a sequence of multiple simple attacks. We therefore recommend that future research on this issue focus on improving the detection methods of complex attacks; in order to provide the system administrator with the opportunity to react promptly, to prevent further damage from attacks.

References

[1] R. Heady, G. Luger, A. Maccabe and M. Servilla, "The architecture of a network-level intrusion detection system Department of Computer Science", College of Engineering, University of New Mexico, **(1990)**.
[2] O. Depren, M. Topallar, E. Anarim and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", Expert Systems with Applications, vol. 29, no. 4, **(2005)**, pp. 713-722.
[3] S. Mukkamala, A. H. Sung and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms", Journal of Network and Computer Applications, vol. 28, no. 2, **(2005)**, pp. 167-182.
[4] Z. Jiong and M. Zulkernine, "A hybrid network intrusion detection technique using random forests", Proceedings of Availability, Reliability and Security, ARES, The First International Conference, **(2006)**.
[5] S. J. Horng, M. Y. Su, Y. H. Chen, T. W. Kao, R. J. Chen, J. L. Lai and C. D. Perkasa, 'A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, vol. 38, no. 1, **(2011)**, pp. 306-313.
[6] C. R. Pereira, R. Y. M. Nakamura, K. A. P. Costa and J. P. Papa, "An Optimum-Path Forest framework for intrusion detection in computer networks", Engineering Applications of Artificial Intelligence, vol. 25, no. 6, **(2012)**, pp. 1226-1234.
[7] M. Govindarajan and R. M. Chandrasekaran, "Intrusion detection using neural based hybrid classification methods", Computer Networks, vol. 55, no. 8, **(2011)**, pp. 1662-1671.
[8] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines", Machine Learning, vol. 46, no. 1-3, **(2002)**, pp. 131-159.
[9] K. P. Wu and S. D. Wang, "Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space", Pattern Recognition, vol. 42, no. 5, **(2009)**, pp. 710-717.

[10] Y. Li, J. L. Wang, Z. H. Tian, T. B. Lu and C. Young, "Building lightweight intrusion detection system using wrapper-based feature selection mechanisms", Computers & Security, vol. 28, no. 6, **(2009)**, pp. 466-475.
[11] S. S. S. Sindhu, S. Geetha and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach",Expert Systems with Applications, vol. 39, no. 1, **(2012)**, pp. 129-141.
[12] P. Louvieris, N. Clewley and X. Liu, "Effects-based feature identification for network intrusion detection", Neurocomputing, vol. 121, no. 0, **(2013)**, pp. 265-273.
[13] C. Guo, Y. J. Zhou, Y. Ping, S. S. Luo, Y. P. Lai and Z. K. Zhang, "Efficient intrusion detection using representative instances", Computers & Security, vol. 39, no. 0, **(2013)**, pp. 255-267.
[14] S. Zargari and D. Voorhis, "Feature Selection in the Corrected KDD-dataset", Proceedings of Emerging Intelligent Data and Web Technologies (EIDWT), Third International Conference, **(2012)**.
[15] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems", Journal of Network and Computer Applications. vol. 34, no. 4, **(2011)**, pp. 1184-1199.
[16] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", Proceedings of Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, **(2009)**.
[17]A. P.Muniyandi, R. Rajeswari and R. Rajaram, "Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm", Procedia Engineering. vol. 30, no. 0, **(2012)**, pp. 174-182.
[18] P. Natesan, P. Balasubramanie and G. Gowrison, "Improving the attack detection rate in network intrusion detection using adaboost algorithm", Journal of Computer Science, vol. 8, no. 7, **(2012)**, pp. 1041-1048.
[19] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Journal of Network and Computer Applications, vol. 30, no. 1, **(2007)**, pp. 114-132.
[20] G. Kim, S. Lee and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection", Expert Systems with Applications, vol. 41, no. 4, Part 2, **(2014)**, pp. 1690-1700.
[21] C. W. Hsu, C. C. Chang and C. J. Lin, "A practical guide to support vector classification", **(2003)**.
[22] J. Juanchaiyaphum, N. Arch-Int, S. Arch-Int and S. Saiyod, "Symbolic data conversion method using the knowledge-based extraction in anomaly intrusion detection system", Journal of Theoretical and Applied Information Technology, vol. 65, no. 3, **(2014)**, pp. 695-701.
[23] E. Rich and K. Knight, Artificial Intelligence McGraw-Hill Education, **(1991)**.
[24] M. A. Hall, "Correlation-based feature selection for machine learning", The University of Waikato, **(1999)**.
[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, **(1967)**.
[26] J. R. Quinlan, C4. 5: programs for machine learning Morgan kaufmann, **(1993)**.
[27] J. R. Quinlan, "Induction of decision trees", Machine learning, vol. 1, no. 1, **(1986)**, pp. 81-106.
[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update", ACM SIGKDD explorations newsletter, vol. 11, no. 1, **(2009)**, pp. 10-18.
[29] S. Hettich and S. D. Bay, "The UCI KD", available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

