Two Layers Multi-class Detection Method for Network Intrusion Detection System

Yali Yuan
Institute of Computer Science
Goldschmidtstr. 7 37077
Goettingen, Germany
yali.yuan@informatik.uni-goettingen.de

Liuwei Huo
School of Computer Science and Engineering
Northeastern University. 110819
Shenyang, China
huoliuwei@163.com

Dieter Hogrefe
Institute of Computer Science
Goldschmidtstr. 7 37077
Goettingen, Germany
hogrefe@informatik.uni-goettingen.de

***Abstract*—Intrusion Detection Systems (IDSs) are powerful systems which monitor and analyze events in order to detect signs of security problems and take action to stop intrusions. In this paper, the Two Layers Multi-class Detection (TLMD) method used together with the C5.0 method and the Naive Bayes algorithm is proposed for adaptive network intrusion detection, which improves the detection rate as well as the false alarm rate. The proposed TLMD algorithm also addresses some difficulties in data mining situations such as handling imbalance datasets, dealing with continuous attributes, and reducing noise in training dataset. We compared the performance of the proposed TLMD method with that of existing algorithms, using the detection rate, accuracy as well as false alarm rate on the KDDcup99 benchmark intrusion detection dataset. The experimental results prove that the proposed TLMD method has a reduced false alarm rate and a good detection rate based on the imbalanced dataset.**

***Index Terms*—Intrusion Detection Systems, C5.0 Method, Naive Bayes Algorithm, Imbalance Data, Detection Rate, False Alarm Rate.**

I. Introduction

Network attack detection is a process of monitoring, detecting, and analyzing unauthorized use, misuse, and abuse of computer systems by both system inside and outside intruders. Network intrusion detection systems play an important role in network information security.

James Anderson firstly gave the notion of intrusion detection in 1980 [2]. Since then different approaches have been proposed to solve network intrusion problems, some of those were presented in [10], [9], [6] and [3]. Generally speaking, IDS can be mainly categorized into two types: misuse-based and anomaly-based. Misuse-based IDSs usually depend on rules written by domain experts, with popular open-source implementations being Snort [19] and Bro [22]. Anomaly-based IDSs should first model all types of normal or valid behaviors, then attempt to detect any type that falls out of normal system operation.

An investigation [1] on latest research literatures presented that designing intelligent intrusion detection model by using data mining and machine learning techniques are popular in IDSs. Decision tree algorithm is one adapted classifier in data mining area, which selects the best split of features with highest classification accuracy. Decision tree algorithms were studied to identify the attacks in IDSs by many researches [13], [5] and [7]. However, there are still challenges in improving the detection accuracy and reducing the high rate of false alarm. ID3 [15] and C4.5 [16] methods are two classical ways to build a decision tree. C5.0 method [18] is a new updated decision tree based on C4.5 method with many new functions. The accuracy and efficiency are better than those of ID3 and C4.5 method [14].

The intrusion detection model based on decision trees mainly includes two types. The first type divides data into normal data and attack data. In [20], authors presented a simple application of decision tree in IDS and gave the normal as well as abnormal detection results. Another new decision tree classifier was proposed by Xiangyang Li et al. [13] by using the different features of raw activity data and different sizes of observation windows in computer network system. The activities were classified as normal or intrusive automatically. They also analyzed the impact of noises in data on the detection performance of the decision tree classifiers. All of methods described above can only categorize the data into normal data and abnormal data and it can not offer any detailed information about attack types. In the second type, multi-class decision trees are established to divide data into many types including normal and several attack types. In paper [12], authors described the process generating the decision tree step by step, and the decision tree was evaluated to get the multi-class detection results. They did experiments by using the KDDcup99 dataset [21] and proved the efficiency of their new decision tree. An effective combined classifier approach by using tree algorithm was proposed by Jasmin Kevric et al. in 2016 [11]. The random tree method and the NBTree algorithm are combined based on the sum rule scheme, which outperforms the individual random tree algorithm. They used the NSL_KDD dataset to evaluate the performance of their detection algorithm.

Although there are many works on this anomaly-based detection systems, several issues are still open and require further research. Different aspects of the problem include multi-class detection accuracy, large scale of traffic data, unbalanced datasets, ambiguous boundaries between normal and intrusive behavior, highly dynamic environments and so on.

Regarding the aforementioned development in this area, the main objective behind our work is to build a new multi-

978-1-5386-1629-1/17/$31.00 ©2017 IEEE

class intrusion detection method by using C5.0 algorithm and Naive Bayes method to improve the detection performance. To evaluate our proposed method, we run various simulations. Ten subsets data are extracted from the training data to reduce the impact of imbalance dataset to the final results and improve the training efficiency. We use the KDDcup99 dataset [21] for our experiments, since it's considered as a benchmark to evaluate the performance of IDSs. Compared with other detection methods, our proposed TLMD algorithm achieves a very low false alarm rate while still maintaining a high detection rate.

The structure of this paper is organized as follows: In section II, the paper describes a brief knowledge background of the proposed TLMD algorithm. The detailed description of the proposed method is presented in section III. Simulation results and performance evaluation are shown in section IV. Finally, section V presents the conclusion and future work.

II. Two Layers of Proposed TLMD Algorithm

A. Selecting Decision Tree Algorithm as First Layer

C5.0 method is developed by Quinlan based on C4.5 algorithm [18]. It has many new technologies except including all the functions of C4.5 algorithm and the most important application is "boosting" technology [8] and [17]. Neural network method, SVM (Support Vector Machine) method, as well as random forest method are very popular in intrusion detection system. Why do we choose C5.0 method as the first layer learning classifier in the proposed TLMD algorithm?

The accuracies and time costs of different algorithms including Neural Network method, SVM algorithm, Random Forest approach and C5.0 algorithm are shown in table I. Table I clearly show that the C5.0 method has a good detection accuracy and a short detection time. We also find that both continuous and categorical features can be handled by C5.0 method. Furthermore, they are robust against redundant and correlated variables, which is important to handle the 41 features of KDDcup99 dataset [21]. Since we are interested in better results and faster approaches, we look out for different algorithms and decide to select C5.0 method. C5.0 method performs faster than ordinary methods and has a good detection accuracy. The detailed advantages of C5.0 method are given below.

- C5.0 method is faster and efficiency.
- C5.0 method can handle the noise and missing data.
- C5.0 method can handle the over fitting and error pruning problem.
- C5.0 method can handle the relevant attributes.
- Both continuous and categorical features can be implemented in a convenient way.

TABLE I
Performance comparison of different algorithms

Method Name	Detection Accuracy(%)	Detection Time(s)
Neural Networks	81.93	598
SVM	83.97	673
Random Forest	92.84	899
C5.0 method	**92.77**	**230**

B. Selecting Naive Bayes Method as Second Layer

A Naive Bayes classifier is a heavily simplified probabilistic based method, which can predict the class membership probabilities [4]. The relationship between independent variable and dependent variable is analyzed to derive a conditional probability for each relationship. The training dataset (TRD) is defined as $\{s_1^{tr}, s_2^{tr}, \cdots, s_i^{tr}, \cdots, s_N^{tr}\}$, where s_i^{tr} is the ith sample in TRD. Each sample s_i^{tr} has M features $F = \{f_1, f_2, \cdots, f_j, \cdots, f_M\}$ and each feature f_j has N feature values $\{f_{j1}, f_{j2}, \cdots, f_{ji}, \cdots, f_{jN}\}$ in TRD. The feature values can be categorical or continuous. The TRD contains H classes $C = \{c_1, c_2, \cdots, c_k, \cdots, c_H\}$. Each sample s_i^{tr} has a specific class label c_k,

For zth sample s_z^{te} in the testing dataset (TED), the trained classifier will predict that s_z^{te} belongs to the class label c_k with the highest posterior probability, conditioned on feature values. That is, the Naive Bayes classifier predicts that the sample s_i belongs to one class label c_{k1}, if and only if it satisfies equation (1),

$$P(c_{k1}|F) > P(c_{k2}|F) \tag{1}$$

where $k1 \neq k2$.

The Bayes theory is shown in equation (2). To compute $P(F|c_k)$ in one dataset with many features is extremely computationally expensive. Thus, in order to simplify computation of evaluating $P(F|c_k)$, Naive Bayes method gives the assumption of class conditional independence. All the attributes are conditionally independent of one another, given the class label of the instance. Thus, in Naive Bayes theory, equation (3) and (4) are given to calculate $P(F|c_k)$.

$$P(c_k|F) = \frac{P(F|c_k)P(c_k)}{P(F)} \tag{2}$$

$$P(F|c_k) = \prod_{j=1}^{m} P(f_j|c_k) \tag{3}$$

$$P(F|c_k) = P(f_1|c_k) \times P(f_2|c_k) \times \cdots \times P(f_m|c_k) \tag{4}$$

III. Implementation of Two Layers Multi-class Detection Method

In this section, we show how to select and build the proposed TLMD method. The general structure of proposed TLMD method is depicted in figure 1.

The proposed TLMD method can be split into the following steps: Firstly, ten subsets are drew from the training dataset according to the predefined rules. Secondly, ten predictors are trained by the given subsets. Then, the final output results of all predictors will be further handled by Naive Bayes method. Lastly, the new sample from the testing dataset will be judged by the TLMD method to obtain its predicted label.

Fig. 1. Structure of proposed TLMD method

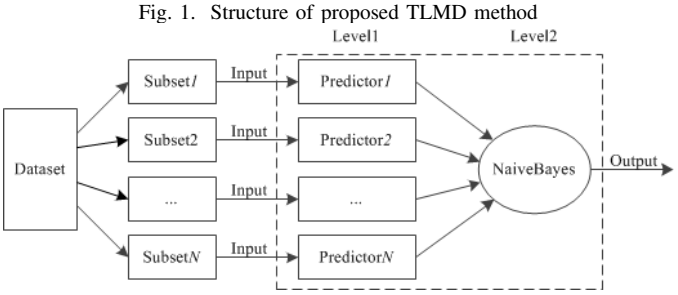


TABLE V
EXAMPLE RESULTS OF TEN PREDICTORS

Feature Name	Proportion(%)
service	100
root_shell	99.94
src_bytes	99.41
dst_bytes	98.36
flag	93.82
land	93.28
dst_host_diff_sry_rate	87.04
count	92.04
dst_host_same_srv_rate	83.79

A. TLMD Algorithm Description

We have a set of possible labels given in equation (5) and we start with training ten predictors by using ten subsets from training dataset first. The results of predictors are described as equation (6),

$$L = \{l_1, l_2, \cdots, l_m\} \tag{5}$$

$$f(pre_1), f(pre_2), \cdots, f(pre_n) \in L \tag{6}$$

where pre_1 is one C5.0 method and $f(pre_1)$ is the predictor result by using pre_1.

The final label can be selected by equation (7).

$$\begin{aligned} & P(l_k|f(pre_1), f(pre_2), \cdots, f(pre_n)) \\ = & \underset{j=1,2,\cdots,m}{\arg\max}\ P(l_j|f(pre_1), f(pre_2), \cdots, f(pre_n)) \end{aligned} \tag{7}$$

However, computing $P(l_j|f(pre_1), f(pre_2), \cdots, f(pre_n))$ directly is difficult and extremely computationally expensive. For this reason, we introduce Naive Bayes method. The final label can be obtained by Naive Bayes method. Finally, if new sample comes, we will use this TLMD algorithm to predict it.

Why do we avoid a majority vote among the predictors? An example for this can be seen in table II. If one predictor can find $R2L$ or $U2R$ attacks and the other nine cannot, then the single predictor which might be right always loses the voting right. Therefore, we introduced Naive Bayes to solve the equation (7).

IV. EXPERIMENTS

A. Datasets Analysis

The given dataset is taken from the annual Knowledge Discovery and Dataset Mining contest (KDDcup99) [21]. All records in the set are simulated tcpdump dataset which is composed from attacks and normal TCP/IP connections. The different attack types fall into four categories:

- **Denial of Service Attack (DoS):** e.g. SYN Flood.
- **Remote to Local Attack (R2L):** e.g buffer overflow to gain root privileges.
- **User to Root Attack (U2R):** e.g. guessing passwords.
- **Probing Attack:** e.g. port scanning.

The different amounts of each attack type in the training dataset are presented in table III. For the testing data, the same information is given in table IV. Note that the 17 boldly listed attack types are new attack types which do not occur in the training dataset. Obviously, we can see that the attack distributions between testing and training dataset are different. Moreover, the tables show that both sets contain about 20% attacks.

B. Datasets Preprocessing

After we take a closer look into the datasets, we decide to completely remove the two constant features is_host_login and $num\_outbound\_cmds$ as we will not gain any information from constant features. To identify the most important features, we use decision trees, since trees choose features with the highest information gain (by entropy) and place the most important features at the top levels of the tree. Therefore, we create ten C5.0 methods and average the usage of the used features to get an idea of the significance for each feature. The reasons why we prefer C5.0 trees over ordinary method are explained in section II. A list of the ten most used features can be seen in table V.

C. Handling imbalance dataset

Table III and IV clearly show that the distributions of each attack type and normal type both in training dataset and testing dataset are greatly different. The distribution of each attack training dataset is given in table III, where the proportion of U2R and R2L is only 0.01% and 0.022%, respectively. Table IV shows the distribution of each attack in testing dataset, where U2R and R2L have many new attack types and they are not in training dataset. How can we handle the imbalance dataset during the training phase? How can we improve the performance of proposed TLMD method during the testing phase?

Since the given dataset consists of 494021 TCP/IP connections, which is rather big and therefore takes a lot of computational time, we decide to use subsets to train the proposed TLMD method instead. We extract ten subsets with replacement. The size of each subset is 23×2000, where 23 includes 22 known attack types and the the normal type from training dataset and each type consists of 2000 connections. Therefore, we strongly change the distribution as some categories like U2R and R2L are very rare. Note that this can gain a better performance on rare categories. Our tests reveal that 2000 samples for each type are a good compromise between efficiency and performance.

TABLE II
EXAMPLE RESULTS OF TEN PREDICTORS

pred1	pred2	pred3	pred4	pred5	pred6	pred7	pred8	pred9	pred10
normal	normal	normal	normal	normal	normal	normal	normal	normal	normal
normal	normal	normal	normal	normal	normal	normal	normal	normal	R2L
R2L	R2L	R2L	R2L	R2L	R2L	R2L	R2L	R2L	R2L
normal	normal	normal	probe	probe	normal	normal	normal	normal	normal
normal	normal	normal	normal	probe	normal	normal	normal	normal	normal
R2L	R2L	normal	normal	R2L	normal	R2L	normal	R2L	R2L
normal	normal	normal	normal	R2L	normal	normal	normal	R2L	R2L
R2L	normal	normal	normal	normal	normal	normal	normal	normal	R2L
normal	probe	DoS	DoS	DoS	DoS	normal	normal	normal	normal
normal	normal	normal	normal	R2L	normal	normal	normal	normal	normal
probe	DoS	normal	probe	probe	probe	normal	normal	normal	normal
probe	normal	DoS	normal	normal	probe	DoS	normal	probe	normal
normal	probe	probe	probe	DoS	probe	probe	normal	normal	probe
probe	probe	probe	probe	normal	normal	probe	probe	probe	normal
normal	normal	normal	normal	R2L	R2L	R2L	normal	normal	normal
probe	DoS	probe	probe	DoS	probe	DoS	probe	DoS	DoS
probe	DoS	DoS	normal	normal	probe	DoS	normal	probe	normal
normal	R2L	R2L	R2L	R2L	R2L	normal	R2L	R2L	normal
probe	probe	probe	probe	normal	normal	normal	probe	normal	normal
probe	probe	probe	probe	probe	probe	probe	probe	probe	probe
				.					
				.					
				.					

TABLE III
PROPORTION OF EACH ATTACK IN THE TRAINING DATASET

Classification of Attacks	Attack Name	Proportion(%)
Probing	Port-sweep, IP-sweep, Nmap, Satan	0.83
DoS	Neptune, Smurf, Pod, Teardrop, Land, Back	79.23
U2R	Buffer-overflow, Load-module, Perl, Rootkit	0.01
R2L	Guess-password, Ftp-write, Imap, Phf, Multihop, spy, warezclient, Warezmaster	0.022

TABLE IV
PROPORTION OF EACH ATTACK IN THE TESTING DATASET

Classification of Attacks	Attack Name	Proportion(%)
Probing	Port-sweep, IP-sweep, Nmap, Satan **Saint**, **Mscan**	1.34
DoS	Neptune, Smurf, Pod, Teardrop, Land, Back **Apache2**, **Udpstorm**, **Processtable**, **Mail-Bomb**	73.90
U2R	Buffer-overflow, Load-module, Rootkit, Perl **Xterm**, **Ps**, **Sqlattack**	0.02
R2L	Guess-password, Ftp-write, Imap, Phf, Multihop, spy, warezclient, Warezmaster, **Snmpgetattack**, **Named**, **Xlock**, **Xsnoop**, **Send-Mail**, **Http-Tunnel**, **Worm**, **Snmp-Guess**	5.26

D. Simulations

1) General Results of Proposed TLMD Algorithm: Four types of network attacks including DoS, Probe, R2L and U2R as well as one normal type are detected in this proposed TLMD method. We use the detection accuracy, detection false alarm rate and detection rate to evaluate the performance of the TLMD method.

We run 30 times of the proposed TLMD method to obtain average results. All simulations are done on a Lenovo G490 laptop, with an Intel(R) core(TM) i5-3230M CPU at 2.60GHz and 6 GB(DDR3 1600MHz) of RAM. We used R version 3.3.1. Confusion matrix for all types is shown in table VI. Performance of proposed TLMD method is given in table VII.

TABLE VI
CONFUSION MATRIX FOR ALL ATTACK TYPE

		Actual				
	Feature	DoS	Normal	Probe	R2L	U2R
Pred	DoS	223717	81	77	0	0
	Normal	5304	60132	197	14089	4
	Probe	254	256	3655	146	0
	R2L	577	88	235	2077	30
	U2R	1	36	2	35	36

E. Comparison with Existing Methods

In order to have a comparison of our method, we compare the proposed TLMD method with SVM algorithm and C5.0 method. The overall accuracies of these three method are shown in table VIII.

The accuracy rate comparison of the proposed TLMD

TABLE VII
PERFORMANCE(%) OF PROPOSED TLMD METHOD

Metrics and Feature	DoS	Normal	Probe	R2L	U2R
Accuracy rate	98.57	95.71	93.77	56.20	75.71
False Alarm Rate	2.023	6.447	0.375	4.887	0.034
Detection Rate	97.33	99.24	87.74	12.71	51.43

TABLE VIII
ACCURACY(%) OF COMPARISON RESULTS

Method Name	Detection Accuracy
SVM	83.97
C5.0	92.77
Propsed TLMD Method	**93.32**

method, SVM algorithm and C5.0 method is given in figure 2. Clearly, the proposed TLMD method outperforms both SVM algorithm and C5.0 method. Figure 3 shows the detection rate comparison of proposed TLMD method, SVM algorithm and C5.0 method while the false alarm rate comparison of these three methods is given in figure 4. It should be observed that the research results of SVM and C5.0 are considerable inferior performance in the aspect of detection capability based on the types of normal, DoS, Probe and R2L. Besides, in figure 2 and 3, we can also see that the performance of the proposed TLMD method is stable and good at the detection capability of all types. As for the false alarm rate given in figure 4. It is clear that the proposed TLMD method has the minimum false alarm value at all types including normal, DoS, Probe, R2L and U2R.

From the aforementioned simulation results, all research methods have less accuracy value and detection value in the aspect of R2L and U2R compared those of normal, DoS and probe attacks. It can be explained as follows: If certain attack types are given only in the testing dataset and not in the training dataset, we define these attack types as new attack types. Due to the fact that in the R2L and U2R categories given in table III and IV, there is a high number of new attack types, which can be considered as the most important reason for the lower performance. Additionally, in the training dataset, the total amount of attacks from the R2L and U2R categories is very low, with 0.01% and 0.022% respectively. In the case of R2L, the relative amount present in the testing dataset is much higher at 5.26%, with half of it being made up of new attack types. The size of each type is relatively large or small, which can deliver more or less impact on the training results.

V. CONCLUSION

In this paper, we have designed a new two layers multi-class detection method for improving the performance of network intrusion detection. C5.0 algorithm and Naive Bayes method are used to build the two layers model. Firstly, the dataset will be preprocessed into ten subsets. Then, we will train all the predictors by using the given subsets. Naive Bayes method is employed to further handle the output of all predictors. KDDcup99 dataset [21], which serves as a reliable benchmark dataset for comparing the performance of

Fig. 2. Detection accuracy comparison of Proposed TLMD method with other detection methods over five types

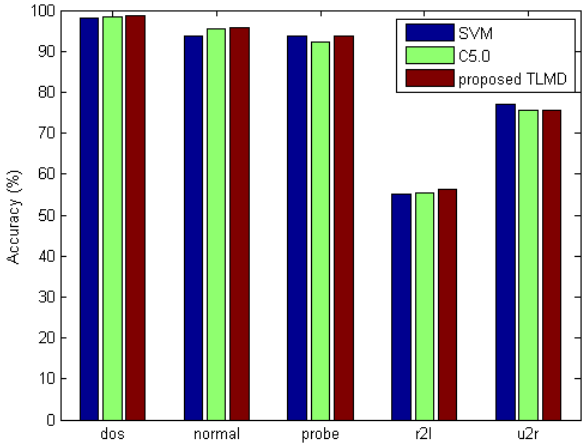
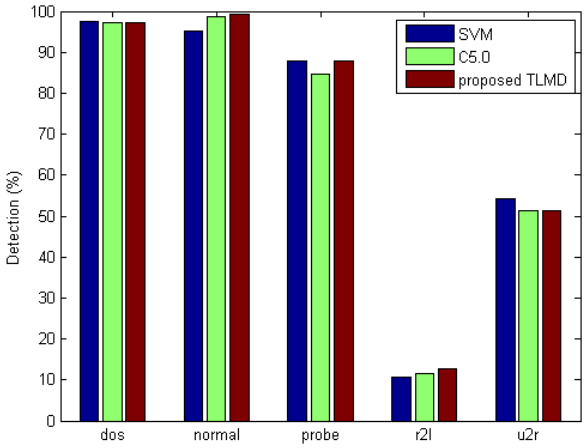


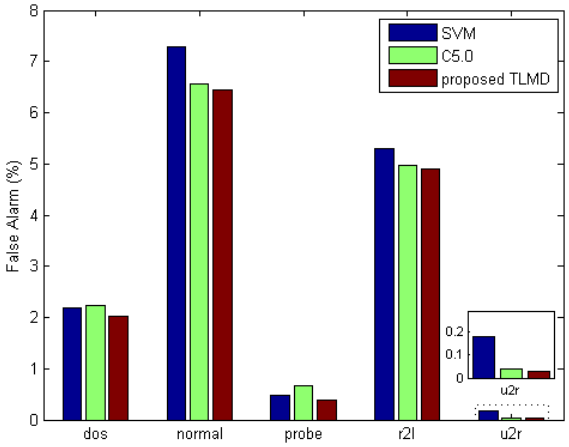
Fig. 3. Detection rate comparison of Proposed TLMD method with other detection methods over five types



network intrusion detection, is employed in the experiments. Ten subsets with replacement which each consisted of 23×2000 connections for all 22 known attack types and one normal type from the training dataset are drew to improve the time efficiency of training phase and reduce the affection of imbalance distributions of different types in training dataset. We compared the proposed TLMD method with SVM and C5.0 based on several important evaluation metrics.

Through the experiments, we show that the proposed multi-class classifier in this paper has superior detection accuracy capability to other methods as a whole and achieves a very low false alarm rate and a high detection rate as well. We run our algorithm for 30 times to obtain the average results and the experiment results prove that our proposed algorithm is reproducible and consistent over a different number of runs.

Fig. 4. Detection false alarm rate comparison of Proposed TLMD method with other detection methods over five types



Theoretical analysis and experiments show that our proposed algorithm achieves a competitive performance, as compared with the other detection algorithms, based on the benchmark dataset.

ACKNOWLEDGMENT

Yali Yuan would like to thank the scholarship support by China Scholarship Council (CSC). Furthermore, the authors gratefully acknowledge Dieter Lechler, Ferdinand Bollwein and Martin Schwarzmaier for their initial contribution for this work. They are all from Institute of Computer Science, Goldschmidtstr. 7 37077, Goettingen, Germany and they are contributed as the co-third author in this paper.

REFERENCES

[1] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[2] J. P. Anderson et al. Computer security threat monitoring and surveillance. Technical report, Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[3] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 378:484–497, 2017.

[4] J. Chen, H. Huang, S. Tian, and Y. Qu. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3):5432–5435, 2009.

[5] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P.-N. Tan. Data mining for network intrusion detection. In *Proc. NSF Workshop on Next Generation Data Mining*, pages 21–30, 2002.

[6] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy. A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, 4(4):753–762, 2013.

[7] 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*, 41(4):1937–1946, 2014.

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

[9] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.

[10] W. Hu, W. Hu, and S. Maybank. Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):577–583, 2008.

[11] J. Kevric, S. Jukic, and A. Subasi. An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications*, pages 1–8, 2016.

[12] J.-H. Lee, J.-H. Lee, S.-G. Sohn, J.-H. Ryu, and T.-M. Chung. Effective value of decision tree with kdd 99 intrusion detection datasets for intrusion detection system. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 2, pages 1170–1175. IEEE, 2008.

[13] X. Li and N. Ye. Decision tree classifiers for computer intrusion detection. *Journal of Parallel and Distributed Computing Practices*, 4(2):179–190, 2001.

[14] R. Pandya and J. Pandya. C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, 117(16), 2015.

[15] J. Quilnan and D. Michie. Discovering rules by induction from large collection of examples. *Expert Systems in the Micro Electronic Age. Edinburgh: Edinburgh UniversityPress*, pages 168–201, 1979.

[16] J. R. Quinlan. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38, 1993.

[17] J. R. Quinlan et al. Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996.

[18] Q. J. R. "c5". *online] http://rulequest.com*, 2007.

[19] M. Roesch et al. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, 1999.

[20] C. Sinclair, L. Pierce, and S. Matzner. An application of machine learning to network intrusion detection. In *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*, pages 371–377. IEEE, 1999.

[21] S. Stolfo et al. The third international knowledge discovery and data mining tools competition. *online] http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html (accessed 20 February 2015)*, 2002.

[22] M. Vallentin, R. Sommer, J. Lee, C. Leres, V. Paxson, and B. Tierney. The nids cluster: Scalable, stateful network intrusion detection on commodity hardware. In *International Workshop on Recent Advances in Intrusion Detection*, pages 107–126. Springer, 2007.