A Two-Stage Classifier Approach using RepTree Algorithm for Network Intrusion Detection

Article *in* International Journal of Advanced Computer Science and Applications · January 2017

DOI: 10.14569/IJACSA.2017.080651

3 authors, including:

Mustapha Belouch
Cadi Ayyad University, Faculty of Sciences and Technologies Marrakech
8 PUBLICATIONS **106** CITATIONS
SEE PROFILE

Mohamed Idhammad
University Ibn Zohr - Agadir
7 PUBLICATIONS **106** CITATIONS
SEE PROFILE

A Two-Stage Classifier Approach using RepTree Algorithm for Network Intrusion Detection

Mustapha Belouch
Laboratory of Applied Mathematics and Informatics, FSTG, Cadi Ayyad University, Marrakesh, Morocco

Salah El Hadaj
National School of Trade and Management Cadi Ayyad University, Marrakesh, Morocco

Mohamed Idhammad
LabSIV, Department of Computer Science, FSA, Ibn Zohr University Agadir, Morocco

***Abstract*—In this paper, we present a two-stage classifier based on RepTree algorithm and protocols subset for network intrusion detection system. To evaluate the performance of our approach, we used the UNSW-NB15 data set and the NSL-KDD data set. In first phase our approach divides the incoming network traffics into three type of protocols TCP, UDP or Other, then classifies into normal or anomaly. In second stage a multiclass algorithm classify the anomaly detected in the first phase to identify the attacks class in order to choose the appropriate intervention. The number of features is reduced from over 40 to less than 20 features, according to the protocol, using feature selection techniques. The detection accuracy of 88,95% and 89,85% was achieved on the complete UNSW-NB15 and NSL-KDD data set, respectively using individual classifier, results are better as compared to the recent work on these data sets.**

***Keywords*—Intrusion detection; REPTree; UNSW-NB15; NSL-KDD**

I. Introduction

The emerging Internet of Things together with the rapid growth of computer networks, connected devices, web applications and cloud computing, highlight now, more than ever, the need for accurate and efficient network security. With the aim to protect confidentiality, integrity and availability against the numerous threats and cyber-attacks, firewalls, authentication methods, intrusion detection and prevention systems have been developed over the years.

An Intrusion Detection System (IDS) is used to identify an unauthorized or malicious action which can compromise the confidentiality, integrity or availability of an information resource [1]. In case of such a detection, the IDS requires the network administrator to intervene. An IDS can be classified based on the type of intrusions that detects with the two primary ones being a misuse intrusion and an anomaly based one. A misuse detection algorithm can only detect known attacks based on the stored intrusion database signature. In an anomaly based detection system, a trained algorithm creates a model of normal activities and activities that deviate from these models are classified as an anomaly [2]. While a misuse or signature based detection is preferred for commercial products due to its high predictability and accuracy, an anomaly detection system is considered as a more effective way to address novel attacks [3].

Unfortunately, modern attacks are continuously changing and enable diverse intrusion mechanisms. The attacks are becoming more intelligent and self-adaptable, able to deal with the current securities of conventional network administrations. This sophistication in threats is very dynamic, which in turn makes it critical for newer security measure adoptions.

An efficient, accurate and real-time IDS is required to present low false positive ratio, high true positive ratio and at the same time entail low detection and response time and maintain a high detection attack rate. Detection response time and overhead are two of the most challenging issues of a modern IDS since computer networks and data information are continuously changing and increasing, making a real-time intrusion detection is a critical feature of a modern IDS [4].

For evaluating the efficiency of an IDS, a modern comprehensive data set that contains contemporary normal and attack activities is required [5]. By analysing an IDSs response to various important outbound and inbound traffic, critical information can be extracted and efficient training of an IDS can be achieved. The NSL-KDD dataset which is an improved version of the original KDDCUP'99 dataset [3], and the UNSW-NB15 is a modern datasets with realistic attacking and normal activities [6].

The proposed anomaly based IDS uses machine learning algorithms for intrusion detection and prediction. Binary and multi-class classification is performed for both normal/attack activities and attack possible states. A true negative state (TN) is considered when the IDS identifies an activity as a normal one with the actual activity being normal. A false positive (FP) case is considered when the IDS identifies an activity as an attack but the actual activity is a normal one. A false positive (FP) is a false alarm, and in a false negative (FN) situation, which is the most critical one, the IDS fails to identify an actual attack.

The remainder of this paper is structured as: Section 2 presents related works currently used in the domain. Section 3 describes the proposed method and the techniques and algorithms used for our approach. Experimental and comparison results are provided in Section 4. Finally, Section 5 provides the final conclusions.

II. Related Work

Since Denning firstly proposed an intrusion detection model [7], many research efforts have been focused on how to effectively and accurately develop most advanced and modern

detection models. Artificial intelligence and machine learning techniques were exploited to identify the underlying patterns and models utilizing training datasets. The most commonly used methods are focused on rule based induction, classification and data clustering.

Based on a previous study [8], many detection algorithms reported high detection rates with relative low false positives. By examining specific datasets such as the KDDCUP'99 one [3], two critical issues can be identified. Firstly, the KDD dataset includes a huge number of redundant records, which causes a significant bias in the learned algorithms towards the most frequent records. Secondly, the difficulty level of the records is quite questionable since about 98% of the records in the train set and 86% of the records in the test set are correctly classified within all the 21 learners.

The new version of KDD dataset, the NSL-KDD, is publicly available and although this dataset still suffers in some of the issues discussed by McHugh [9] it can be considered as an adequate benchmark for evaluating intrusion detection methods. A standard KDDTrain+ and KDDTest+ is presented in [3], in order to train and test algorithms in such a way that researchers can easily compare their results. In this work, we compare our proposed method with other two state-of-the-art methods using the same and complete dataset.

In [6], a recent dataset, the UNSW-NB15, includes real-world normal and abnormal network traffic in a synthetic environment. This dataset was utilized in [5] for statistical and evaluation purposes by comparing five different algorithms DT, LR, NB, ANN, and EM clustering, for measuring their performance in terms of accuracy and False Alarm Rate (FAR) against the KDD99 dataset. The evaluation results showed that the DT technique achieved the best efficiency. Furthermore, the results of the two datasets were also compared showing that the efficiency techniques using the KDD99 data set were better than when using the UNSW-NB15 dataset. As a consequence, the UNSW-NB15 dataset can be considered more complex and a better representative of the modern attack and normal network traffic, making it more appropriate for the evaluation of existing and the proposed NIDS methods.

Many researchers have chosen the NSL-KDD dataset since it is publicly available. A binary classifier is used in [4], [10]–[14] to identify the incoming network traffic as normal or attack. On the other side, some works propose a multi-class classifier for classifying the incoming network traffic into five categories; normal, DoS, U2R, R2L or Probe [15]–[24]. In [12], [17], [20], [23], [25], researchers used a random portion of the NSL-KDD for the training and testing dataset but without clearly denoting which subpart of the NSL-KDD was used (KDDTrain+ or KDDTest+). As a result, the comparison of these methods is inefficient because of the different used datasets for both the training and testing. Other works report the use of the same datasets for training and testing allowing a high accuracy results from 94,7% to 99,7% [12], [26].

The work in [10] utilizes a fuzzy classification over the NSL-KDD using the KDDTrain+ and KDDTest+ for training and testing, respectively, without discrimination of the attack types, with an overall achieved accuracy of 82,74%. In [27], three types of detection agents were generated according to TCP, UDP and ICMP protocols with a reported accuracy rate at 91.21% on the KDDCUP'99 dataset. After preprocessing, 32 attributes for the TCP detection agent were selected, 21 attributes for the UDP detection agent and 18 attributes for the ICMP detection agent were chosen. The training time of the proposed method was 194 seconds. In [13], the NSL-KDD was divided to TCP, UDP and ICMP and feature selection was applied, with a reported low accuracy only in the UDP subset.

A number of researchers have employed different feature selection techniques to reduce the number of features and to eliminate irrelevant features from the NSL-KDD data set. The work in [13] combines information gain and a genetic algorithm for selecting 17, 10 and 5 features for the TCP, UDP and ICMP, respectively. A deep belief network, a gain ration and a chi-square were used to select only 13 features based on the proposed work in [15]. Principal component analysis was also used in [19] to reduce the selected features to 23. The authors in [24] apply a combining classifier with NBTree and RandomTree algorithm in the NSL-KDD dataset for detecting the normal and attack traffic with an achieved accuracy of 89,24% along 41 attributes. In [5], a different multiclass classifier is applied to identify normal traffic or nine attack types: Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode and Worm. The highest result of 85,56% was achieved using a Decision Tree algorithm with all the attributes.

III. Proposed Research Method

The proposed method introduces a novel approach of model creation for better results in terms of accuracy and training time using individual classifiers instead of combining multiple algorithms. The system architecture is explained throughout this Section along with a the data preprocessing techniques and the classification algorithm. A detailed analysis of the NSL-KDD and UNSW-NB15 datasets that were exploited in the training and testing of the proposed model are presented in the experimental section.

A. Proposed Architecture

The architecture of the proposed approach is shown in Fig. 1 where the intrusion detection model is based on two main stages and on a Reduced Error Pruning Tree (REPTree) algorithm for classification and identification of the incoming network traffic.

In the first stage, the incoming traffic flow is firstly classified upon its protocol as TCP, UDP and other. The reason for such protocol classification lays on the different protocol formats which subsequently defines the different needed features for each one. Data pre-processing is applied to each of the three subsets to eliminate any unrelated features and noisy outliers. The network traffic is defined as normal or attack in this stage, in order to speed up the control of the network.

In the second stage, a pre-trained multiclass classifier is launched whenever an attack was identified by the first classifier for identifying the attack type and providing the appropriate response.

In a network traffic dataset the distribution of connections of various protocols is not even. Since connections in some protocols are more frequent against others, this protocol imbalance affects the pre-processing. The proposed TCP, UDP and

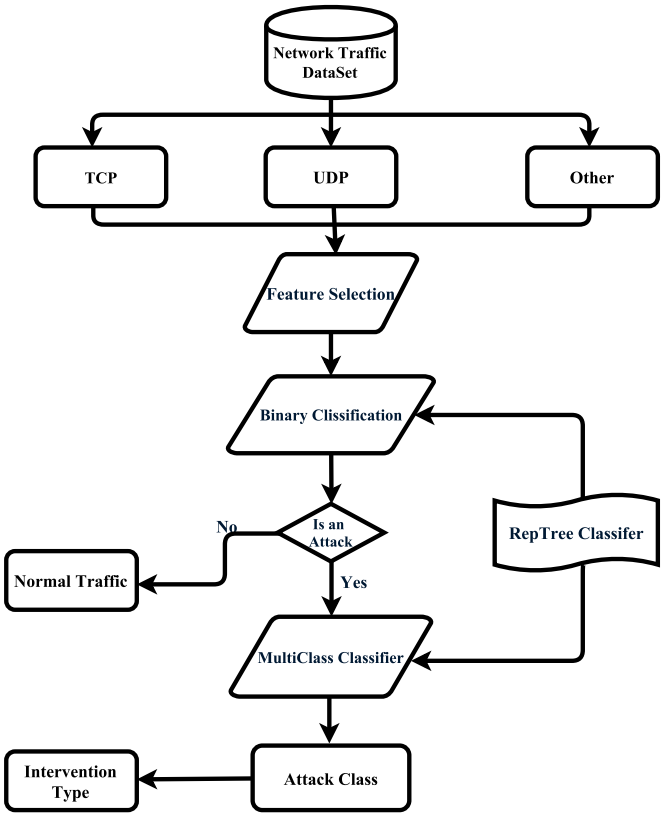

Fig. 1. Proposed architecture.

Other protocol discrimination reduces this effect of protocol imbalance in the dataset.

Feature selection is used also to reduce the size of the analyzed dataset through deleting the unrelated, redundant or irrelevant features. The combination of information gain and consistency through an evolutionary search method was used for the proposed feature selection. Since attributes are filtered by measuring the information gain with respect to the class, the required resources for dataset tuple classification are minimized. The information required conveyed by a tuple of a probability distribution D is given by:

$$Info(D) = -\sum_{i=1}^{\infty} p_i log(p_i) \quad (1)$$

Where, P_i is the probability that an arbitrary tuple D belongs to a class C_i and Info (D) is the entropy of the tuple in D. If we partition a set of samples T on the basis of a non-categorical attribute X into sets $T_1, T_2, \ldots T_m$, then the information needed to identify the class of an element of T is given by:

$$Info(X,T) = \sum_{i=1}^{m} \frac{|T_i|}{|T|} \times Info(T_i) \quad (2)$$

The information gain, $Gain(X,T)$, is then defined as:

$$Gain(X,T) = Info(T) - Info(X,T) \quad (3)$$

TABLE II. DISTRIBUTION OF ATTACKS IN EACH PROTOCOL ON DATASETS

Data set		CLASS	TCP	UDP	Other	Total
NSL-KDD	Training	DoS	42,188	892	2,847	58,630
		Probe	5,857	1,664	4,135	
		U2R	49	3	0	
		R2L	995	0	0	
	Testing	Dos	6,739	14	706	12,832
		Probe	1,864	317	240	
		U2L	67	0	0	
		L2R	2,367	514	4	
UNSW-NB15	Training	Backdoor	272	28	1,446	83,341
		Analysis	564	0	1,436	
		Fuzzers	11,761	4,945	1,478	
		Shellcode	557	576	0	
		Reconnaissance	5,100	3,586	1,805	
		Exploits	19,689	624	13,080	
		DoS	2,281	358	9,625	
		Worms	115	15	0	
		Generic	486	39,229	285	
	Testing	Backdoor	51	6	526	45,332
		Analysis	58	0	619	
		Fuzzers	3,713	1,098	1251	
		Shellcode	193	185	0	
		Reconnaissance	1,865	1,304	327	
		Exploits	7,754	250	3,128	
		DoS	1,055	169	2,865	
		Worms	38	6	0	
		Generic	520	18,303	48	

A ranker algorithm ranks the features in the data set based on their redundancy and relevancy and allowed us to select the appropriate number of features based on our requirements.

To evaluate the performance we used the 10-fold cross validation technique. After randomly dividing the training dataset into 10 distinct parts, the model is trained with 9 parts and one part is selected for testing in each iteration. The value of 10 was chosen empirically due to its adequate performance in estimation error, low bias, low overfitting and low variance.

B. Reduced Error Pruning Tree

Decision tree classifiers like ID3, C4.5, CART, build a decision tree model based on instances of the training dataset. The root and the internal nodes in a decision tree represent the attributes and the leaf nodes represent the classes. However, a decision tree classifier can generate large decision trees that are overfitted to the training set. This effect limits the performance of the classifier and requires more resources in terms of memory allocation. This issue was solved by optimizing the size of the decision tree after applying pruning. The pruning, known also as reduced error pruning, was achieved by the method proposed by Quinlan [28]. While traversing the internal nodes from downwards to upwards, a procedure that checks and replaces each internal node with the most frequent class is initiated, without affecting the trees accuracy. The procedure continues pruning the nodes until any further pruning would decrease the accuracy.

REPTree is considered a fast decision tree learner which builds a decision/regression tree using information gain as the splitting criterion, and prunes it using the reduced error pruning method. Reduced Error Pruning results in a more accurate and simple classification tree, even in cases with large amount of training and testing data.

IV. EXPERIMENTS AND RESULTS

To evaluate the performance of our proposed two stage classifier, a series of experiments on the NSL-KDD and the

TABLE I. NORMAL / ATTACK NSL-KDD AND UNSW-NB15

Data Set		TCP		UDP		Other		Total
		Normal	Attack	Normal	Attack	Normal	Attack	
NSL-KDD	Training	53,600	49,089	12,434	2,559	1,309	6,982	125,973
	Testing	7,842	11,038	1,776	845	93	950	22,544
UNSW-NB15	Training	39,121	40,825	13,922	49,361	2,957	29,155	175,341
	Testing	27,848	15,247	8,097	21,321	1,055	8,764	82,332

TABLE III. THE SELECTED FEATURES FOR EACH PROTOCOL

Data set	Protocol	Selected Features	No. of selected features
NSL-KDD	TCP	2,4,5,32,33,34,36,37,39	9
	UDP	2,4,5,7,22,28,29,32,33,34,35,39	12
	Other	2,34,5,28,32	6
UNSW-NB15	TCP	1,3,5,7,8,12,16,19,21,22,24,26,27, 28,30,31,34,35,36,40	20
	UDP	6,7,8,9,10,11,12,13,17,27,32,35	12
	Other	6,7,8,9,10,11,12,13,17,27,32,35	12

TABLE VIII. PERFORMANCE COMPARISON ON NSL-KDD

Classifier	Accuracy	Train (s)	Test (s)
NBTree+RandomTree [24]	89.24	50.29	0.93
REPTree	**89.85**	**1.17**	**0.24**

TABLE IX. PERFORMANCE COMPARISON ON UNSW-NB15

Classifier	Accuracy	Train (s)	Test (s)
Decision Tree [5]	85.56	7.66	0.84
REPTree	**88.95**	**2.69**	**0.37**

UNSW-NB15 dataset were performed. In these experiments, we implemented and evaluated the proposed methods in the Weka data mining software on a 2.5 GHz Intel Core i5 CPU with 4 GB RAM.

The NSL-KDD dataset includes 41 features with normal classes and 4 types of attacks: Probe, R2L, U2R and Denial of Service Attack (DoS) [3]. The generated datasets, KDDTrain+ and KDDTest+ include 125,973 and 22,544 instances, respectively. During the performance evaluation of the first classifier we used binary class labels (normal or attack) as shown in Table I, where for the second classifier we selected only attack-type labeled classes, as shown in Table II.

The UNSW-NB 15 dataset involves nine attack categories and 49 features [6]. This dataset was divided into 175,341 and 82,332 records for training and testing, as shown in Tables I and II, respectively.

Both used datasets NSL-KDD and UNSW-NB 15 are publicly available, the volume and distribution of the training and testing dataset are presented in Table I for the binary classification and in Table II for the multi-class classification for each protocol.

For the comparison results, we employed also four different learning algorithms for the training and testing dataset in order to compare them against the REPTree classifier. We used all 40 features of the NSL-KDD dataset and 42 features of the UNSW-NB15, where the overall accuracy and the performance of the classification is expressed in terms of precision and training time, respectively. The selected features are given in Table III for each protocol.

An assessment was also performed utilizing a reduced dataset (KDDTrain+ and KDDTest+ datasets from the NSL-KDD and UNSW-NB15) since computational speed is essential for IDS systems running on routers and network appliances. The training and testing was conducted using the reduced feature set shown in Table III for each dataset and for each protocol. The features were selected based on the information gain feature ranking and consistency through an evolutionary search method. The results are presented in Tables IV to VII where the classification accuracy for the selected features is proved to be better when compared with the all features approach.

Furthermore, we performed four experimental series over the two datasets based on the selected features of Table III. We also compared the results with decision tree, neural network, nave Bayes and random tree approaches. The average values of the results are shown in Tables IV to VII and the respective comparison in terms of accuracy, train and test time are shown in Tables VIII and IX.

In the case of binary classification, using the UNSW-NB15 dataset of Table IV, the REPTree algorithm performed the best with the nave Bayes presenting the worst performance. The highest detection accuracy was achieved on the Other protocols and low accuracy achieved on TCP protocol. Similar results are obtained with the NSL-KDD dataset in Table V with the high detection rate evident on the Other protocols and the lowest on UDP protocol.

For the multi-class classification results on the UNSW-NB15 dataset, Table VI shows a better accuracy with decision for REPTree and lower accuracy when using nave Bayes or neural networks and prediction was very difficult on other protocols. The efficiency of detection is quite high in UDP and substantially lower on TCP protocol. Table VII shows results on the NSL-KDD dataset, REPTree algorithm achieved the best accuracy detection and prediction was difficult on UDP protocol.

Table VIII shows the comparison results of the proposed model against the combined method of Random Tree and NBTree classifiers [24]. The results show a quite same accuracy performance but with the advantage of better training and testing time performance. Table IX shows the comparison of results for the UNSW-NB15 dataset with the best accuracy obtained in [5]. Our model presents the best performance in terms of accuracy and training and testing time.

V. CONCLUSION

In this paper, we proposed a two stage classification network intrusion detection system based on the REPTree algorithm. The NSL-KDD dataset and UNSW-NB15 dataset were used to evaluate the performance of our novel detection algorithm. Network traffic if firstly divided into different classes according to the different network protocol. In the first stage we classify the incoming network traffic into normal or attack classes. In case of attack traffic, the second classifier identifies the type of the attack for providing the best necessary

TABLE IV. BINARY CLASSIFICATION ON UNSW-NB15

UNSW-NB15	All Features				Feature selection			
N/A	TCP	UDP	Other	Accuracy	TCP	UDP	Others	Accuracy
DT	81.13	89.30	98.58	86.13	82.01	92.17	99.69	87.74
ANN	84.13	85.06	99.66	86.31	71.96	90.03	99.30	81.67
NB	70.64	87.34	99.50	80.04	71.90	87.82	99.69	80.90
RandomTree	81.35	90.25	98.64	86.59	81.17	91.77	99.42	87.13
RepTree	83.48	90.12	99.85	87.80	**84.48**	**91.88**	**99.85**	**88.95**

TABLE V. BINARY CLASSIFICATION ON NSL-KDD

NSL-KDD	All Features				Feature Selection			
N/A	TCP	UDP	Other	Accuracy	TCP	UDP	Others	Accuracy
DT	85.47	84.5	94.82	85.78	78.67	84.50	94.82	80.09
ANN	82.59	52.68	94.82	79.67	75.28	55.32	93.95	73.82
NB	78.51	72.22	95.11	78.54	72.43	73.29	95.11	73.57
RandomTree	78.41	84.50	95.68	79.91	83.25	73.52	96.26	82.72
RepTree	87.07	85.73	95.20	87.29	**90.02**	**85.76**	**97.12**	**89.85**

TABLE VI. MULTI-CLASS CLASSIFICATION ON UNSW-NB15

UNSW-NB15	All Features				Feature Selection			
Attacks	TCP	UDP	Other	Accuracy	TCP	UDP	Other	Accuracy
DT	81.03	98.27	27.21	78.73	85.13	98.13	35.69	81.68
ANN	77.24	96.30	35.54	78.14	70.74	95.52	35.32	75.54
NB	83.60	93.68	8.70	73.86	76.00	97.13	32.56	77.53
RandomTree	83.76	93.40	21.30	76.21	79.11	98.04	31.16	78.74
RepTree	81.39	97.97	29.74	79.20	**84.46**	**98.11**	**34.81**	**81.28**

TABLE VII. MULTI-CLASS CLASSIFICATION ON NSL-KDD

NSL-KDD	All Feature				Feature Selection			
Attacks	TCP	UDP	Other	Accuracy	TCP	UDP	Other	Accuracy
DT	79.77	38.93	99.57	78.54	74.09	38.93	99.57	73.66
ANN	73.32	38.93	99.57	72.99	79.44	38.93	99.57	78.26
NB	71.95	38.69	99.05	71.76	77.48	38.69	99.05	76.52
RandomTree	69.39	38.46	98.63	69.51	77.68	38.46	99.57	76.71
RepTree	71.43	38.93	99.57	71.37	**85.64**	**38.93**	**99.57**	**83.59**

response. Extensive evaluation and comparison results showed that the proposed two stage classifier model yields better results in terms of speed of detection and prediction accuracy rate.

Attacks classification experiments on both NSL-KDD and UNSW-NB15 are still not perfect especially for UDP and Other protocols. In future work, we will improve the detection accuracy in these protocols.

REFERENCES

[1] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Sri- vastava, P.-N. Tan, *Data mining for network intrusion detection*, in: Proc. NSF Workshop on Next Generation Data Mining, 2002, pp. 21-30.

[2] A. K. Ghosh, J. Wanken, F. Charron, *Detecting anomalous and unknown intrusions against programs*. in: Proc. Computer Security Applications Conference, 1998, pp. 259-267.

[3] M.Tavallaee,E.Bagheri,W.Lu,A.-A.Ghorbani, *A detailed analysis of the KDD CUP 99 data set.* In : Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on. IEEE, 2009. p. 1-6.

[4] H. M. Harb, A. S. Desuky, *Adaboost ensemble with genetic algorithm post optimization for intrusion detection*. Update 2 (2011) 1.

[5] N. Moustafa, J. Slay, *The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set.* Information Security Journal: A Global Perspective (2016).

[6] N. Moustafa, J. Slay, *Unsw-nb15: a comprehensive data set for network intrusion detection systems (UNSW- NB15 network data set)*. in: Military Communications and Information Systems Conference, 2015, pp. 1-6.

[7] D. E. Denning, *An intrusion detection model*, IEEE Transactions on software engineering (2) (1987) 222-232.

[8] S.X.Wu,W.Banzhaf, *The use of computational intelligence in intrusion detection systems: A review*.Applied Soft Computing 10 (1) (2010) 1-35.

[9] J. McHugh, *Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory*. ACM Transactions on Information and System Security 3 (4) (2000) 262-294.

[10] P. Krömer, J. Platoš, V. Snáš, A. Abraham, *Fuzzy classification by evolutionary algorithms*. in: IEEE International Conference on Systems, Man, and Cybernetics, 2011, pp. 313-318.

[11] M. Panda, A. Abraham, M. R. Patra, *Discriminative multinomial naive bayes for network intrusion detection*. in: International Conference on Information Assurance and Security, 2010, pp. 5-10.

[12] M. Panda, A. Abraham, M. R. Patra, *A hybrid intelligent approach for network intrusion detection*. Procedia Engineering 30 (2012) 1-9.

[13] S. Sethuramalingam, E. Naganathan, *Hybrid feature selection for network intrusion*. International Journal on Computer Science and Engineering 3 (5) (2011) 1773-1780.

[14] J. Koshal, M. Bag, *Cascading of C4.5 decision tree and support vector machine for rule based intrusion detection system*. International Journal of Computer Network and Information Security 4 (8) (2012) 8.

[15] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, A. E. Hassanien, *Hybrid intelligent intrusion detection scheme*. in: Soft computing in industrial applications, 2011, pp. 293-303.

[16] P. Natesan, P. Rajesh, *Cascaded classifier approach based on adaboost to increase detection rate of rare network attack categories*. in: International Conference on Recent Trends In Information Technology, 2012, pp. 417-422.

[17] R. S. Naoum, N. A. Abid, Z. N. Al-Sultani, *An enhanced resilient back-propagation artificial neural network for intrusion detection system*. International Journal of Computer Science and Network Security 12 (3) (2012) 11.

[18] R. S. Naoum, Z. N. Al-Sultani, *Learning vector quantization (LVQ) and k-nearest neighbor for intrusion classification*. World of Computer Science and Information Technology Journal 2 (3) (2012) 105-109.

[19] F. E. Heba, A. Darwish, A. E. Hassanien, A. Abraham, *Principle components analysis and support vector machine based intrusion detection system*. in: International Conference on Intelligent Systems Design and Applications, 2010, pp. 363-367.

[20] Y. Zhang, L. Wang, W. Sun, R. C. Green II, M. Alam, *Distributed intrusion detection system in a multi-layer network architecture of smart grids*. IEEE Transactions on Smart Grid 2 (4) (2011) 796-808.

[21] S.Joseph,etal., *Feature reduction using principal component analysis for effective anomaly-based intrusion detection on nsl-kdd*. International Journal of Engineer- ing Science and Technology 1 (2) 1790-1799.

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

[23] S. Kumar, S. Nandi, S. Biswas, *Research and application of one class small hypersphere support vector machine for network anomaly detection*. in: International Conference on Communication Systems and Networks, 2011, pp. 1-4.

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

[25] C.R.Pereira,R.Y.Nakamura,K.A.Costa,J.P.Papa, *An optimum-path forest framework for intrusion detection in computer networks*. Engineering Applications of Artificial Intelligence 25 (6) (2012) 1226-1234.

[26] G. MeeraGandhi, K. Appavoo, S. Srivasta, *Effective network intrusion detection using classifiers decision trees and decision rules*. International Journal of Advanced Network and Application 2.

[27] S. Teng, H. Du, N. Wu, W. Zhang, J. Su, *A cooperative network intrusion detection based on fuzzy SVMs*. Journal of Networks 5 (4) (2010) 475.

[28] J. R. Quinlan, *Simplifying decision trees*. International journal of man-machine studies 27 (3) (1987) 221-234.

[29] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.