Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 2199-2206
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.5369

Intrusion detection in computer networks using a cost sensitive ensemble classifier

Alaa Thamer Mahmood[a,*], Raed Kamil Naser[b]

[a] Technical Instructors Training Institute, Middle Technical University, Baghdad, Iraq
[b] Administration Directorate, Ministry of Defense, Baghdad, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

The growing use of Internet technology and the attack on computer networks have made intrusion detection systems an essential part of computer security. Conventional intrusion control methods such as firewalls or access control systems are no longer alone able to withstand attacks. Therefore, the need to detect new attacks and anomalies is inevitable. The dataset used in this paper is called NSL-KDD which includes 5 classes: one of them is normal and the other four classes are attacks. In the presented work, an ensemble classifier based on the mean probability of attacks is adopted. The true detection rate of the proposed system is 99.89% which is more than other competing methods. Moreover, the ensemble classifier achieved an F1-measure of 92.48%. To improve the F1 measure, we used a meta-classifier called meta-cost which incorporates a cost matrix to transform the original classifier into a cost-sensitive classifier. By this idea, we achieved an F1-measure of 94.1% which outperforms than non-cost sensitive ensemble classifier. These results show that the proposed system can be used as a suitable defence tool to detect intrusion against cyber-attacks.

Keywords: Intrusion detection, Classification, Ensemble classifier, Machine learning, Computer networks.
2010 MSC: Please write mathematics subject classification of your paper here.

1. Introduction

Intrusion detection system (IDS) is any software, hardware, or combination of both, which is responsible for blocking attackers or hackers when they are trying to enter a computerized system or

*Corresponding author
Email addresses: alaa.thamer@mtu.edu.iq (Alaa Thamer Mahmood), ayhar_2013@yahoo.com (Raed Kamil Naser)

Received: May 2021 *Accepted:* June 2021

computer network [10]. These detection systems generally consist of 5 parts including information gathering, system investigation, logging information, control and management, and data analysis part [5, 6]. The main part of an IDS is data analysis that is divided into 3 phases: designing the analysis engine, analysis, and modification [2].

Most of modern methods in IDS use machine learning techniques in analysis phase. the most challenging issue for using such techniques is defining discriminative features and selecting the best model (i.e. classifier). Chen et al. [3] adopted support vector machine (SVM) and Neural Network (NN) to classify normal and attacks. The results showed that SVM outperforms than NN. In another study [9], Sarasamma et al. demonstrated that different categories of features are effective for different types of attack. Hence, they partitioned the $KDD99$ dataset into 3 categories. Afterwards, they utilized a Self-Organizing Map (SOM) with 3 layers for classification. Muda et al. [7] combined Naive Bayes and $K-means$ to improve the detection rate of their proposed IDS. Based on the results, they achieved a precision and recall of 99.6% and 99.8% on $KDD99$, respectively. In [1], 6 features were selected among 41 features of $KDD99$ dataset based on etropy of features. Based on the experiments, they achieved an accuracy of 97.25%. Muniyandi et al. [8] cascaded $K-means$ clustering and $C4.5$ decision tree classifier to improve the results. In their presented work, the entire dataset is clustered into K partitions. Then, $C4.5$ desicion tree is trained on each cluster. In [11], Chi-squared method has been combined with multi-class SVM for an IDS. Data in real world usually sufferes from noise, and IDS works in such noisy environments. Therefore, noisy data degrades the performance of an intrusion detection system. Hussain and Lalmuanawma [4] compare the performance of different classifiers on a noisy dataset. The results demonstrated that a neural self-organizing map is more robust against noise.

To the best of our knowledge, none of the state of the art methods did not design a cost sensitive solution for intrusion detection. In the presented study, we incorporated the concept of cost matrix into an ensemble classifier to achieve a cost sensitive accurate classifier. The rest of this paper is organized as follow: section 2 describes the dataset and the details of the proposed method. The experimental results are analyzed in section 3. Finally, section 4 concludes the paper.

2. Materials and Methods

The proposed method for intrusion detection consists of 6 main steps: dataset preparation, base classifier selection, combining base classifiers, incorporating cost matrix, making the ensemble classifier cost sensitive, and evaluation. The block diagram of the proposed method is illustrated in Figure 1. All the mentioned steps are described in details as follows:

Step 1 (dataset preparation):
We utilized a standard available dataset called NSL-KDD to train and test the proposed ensemble classifier. The prepared samples of the dataset have been tagged by specialists. The NSL-KDD dataset consists of 41 features and 5 classes (i.e. one normal class and 4 attacks including Dos, $U2R$, $R2L$, and $Prob$). The details of NSL-KDD dataset such as the type of attack and the number of samples for each class are summarized in Table 1.

Step 2 (base classifier selection):
Some well-known classifiers such as Random Commitee, PART, Random Subspace, Bagging, IBK, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Logit Boost, and AdaBoost are selected for evaluation to determine the base classifiers. The classifiers with higher accuracy are

Table 1: Details of NSL-KDD dataset

Attacks	Type Of Attacks	Count	Total
Normal	Normal	67343	67343
Dos	Back	956	45927
	Land	18	
	Neptune	41214	
	Pod	201	
	Smurf	2646	
	Teardrop	892	
Probe	Ipsweep	3599	11656
	Portsweep	2931	
	Satan	3633	
	Nmap	1493	
R2L	ftp_write	8	995
	Guess_passwd	53	
	Imap	11	
	Multihop	7	
	Phf	4	
	Warezmaster	20	
	Spy	2	
	Warezclient	890	
U2R	Buffer_overflow	30	52
	Loadmodule	9	
	Perl	3	
	rootkit	10	

combined to construct the ensemble classifier.

Step 3 (combining base classifiers):
The Ensemble classifiers combine different individual classifiers called base classifiers. Each base classifier is a single classifier which is trained on a training set. Then, the trained classifier with its parameters and hyper-parameters are saved as a base classifier. To combine the outputs of the base classifiers, the majority vote strategy has been used to construct the ensemble classifier. In fact, Instead of constructing a single powerful classifier with many parameters, we combine simple base classifier to achieve a powerful ensemble in which the performance of the ensemble is higher than the performance of the base classifiers individually.

Step 4 (incorporating cost matrix):
The cost matrix is implemented using Cost-Sensitive-Classifier method on the most accurate evaluated ensemble classifier. In the presented study, the cost matrix is considered for the attack classes.

Table 2: Confusion Matrix

Prediction			
Intrusion	Normal		
b	a	Normal	Actual
d	c	Intrusion	

The normal class has no classification cost.

Step 5 (making the ensemble classifier cost sensitive):
To make the ensemble classifier cost sensitive, a new weighting strategy for cost matrix has been proposed. In this way, the cost of each class depends on the number of misclassified samples for the corresponding class. To this end, the maximum number of misclassified samples for each attack is calculated and then it is analized separately to find a proper weight leading to a good performance.

Step 6 (evaluation):
To calculate the overall performance of the ensemble classifier, 10-fold cross validation is adopted. Based on this strategy, the original dataset is randomly partitioned into 10 equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

3. Results and discussion

As mentioned in the previous sections, an intrusion detection system can be regarded as a classifier. Therefore, conventional classification performance measures such as accuracy, precision, recall, and F1-measure are useful to evaluate such systems. These measures are calculated from a confusion matrix (Figure 2) as bellow:

$$Accuracy = \frac{a+d}{a+b+c+d} \tag{3.1}$$

$$Recall = \frac{d}{d+c} \tag{3.2}$$

$$Percision = \frac{d}{d+b} \tag{3.3}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.4}$$

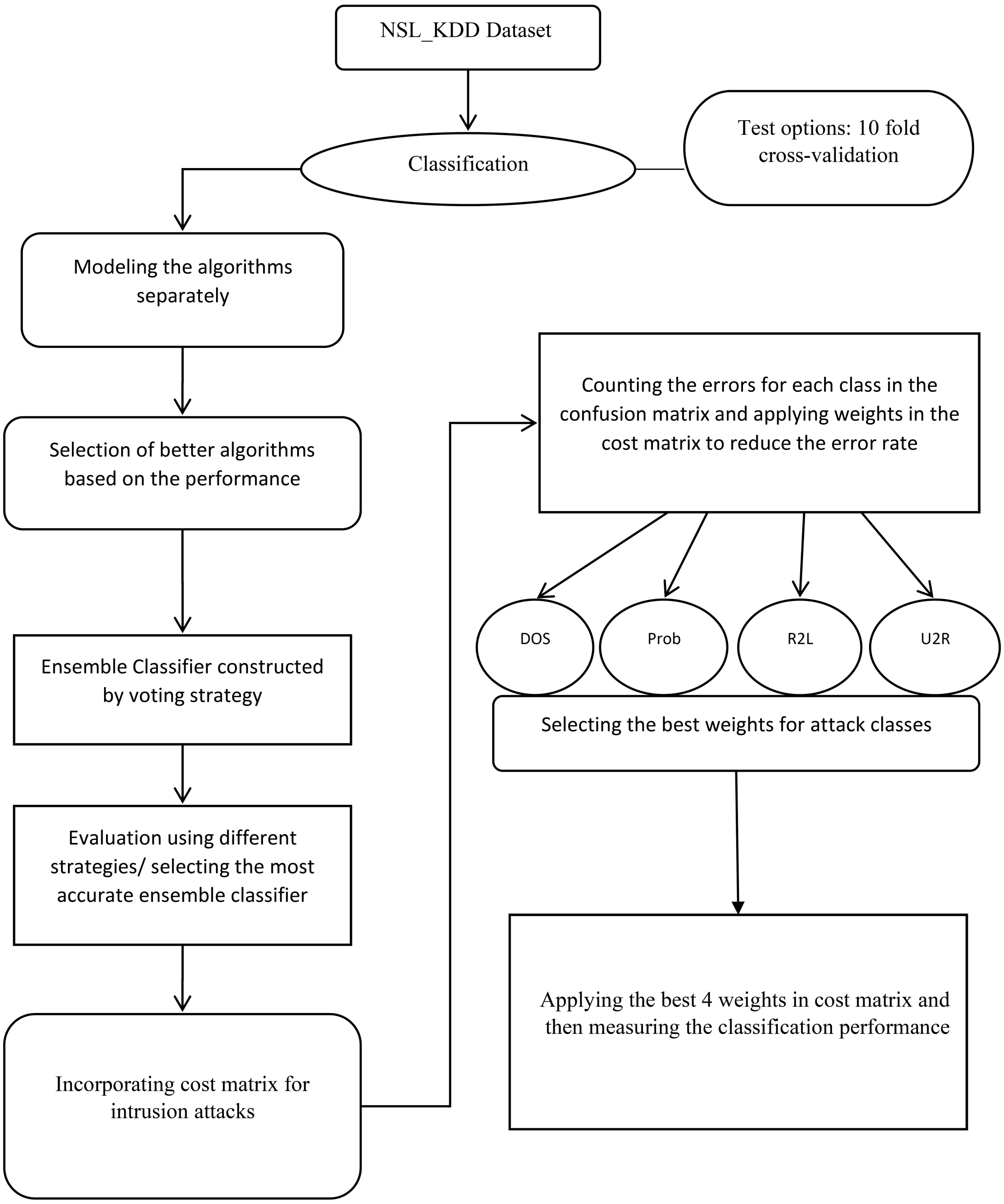
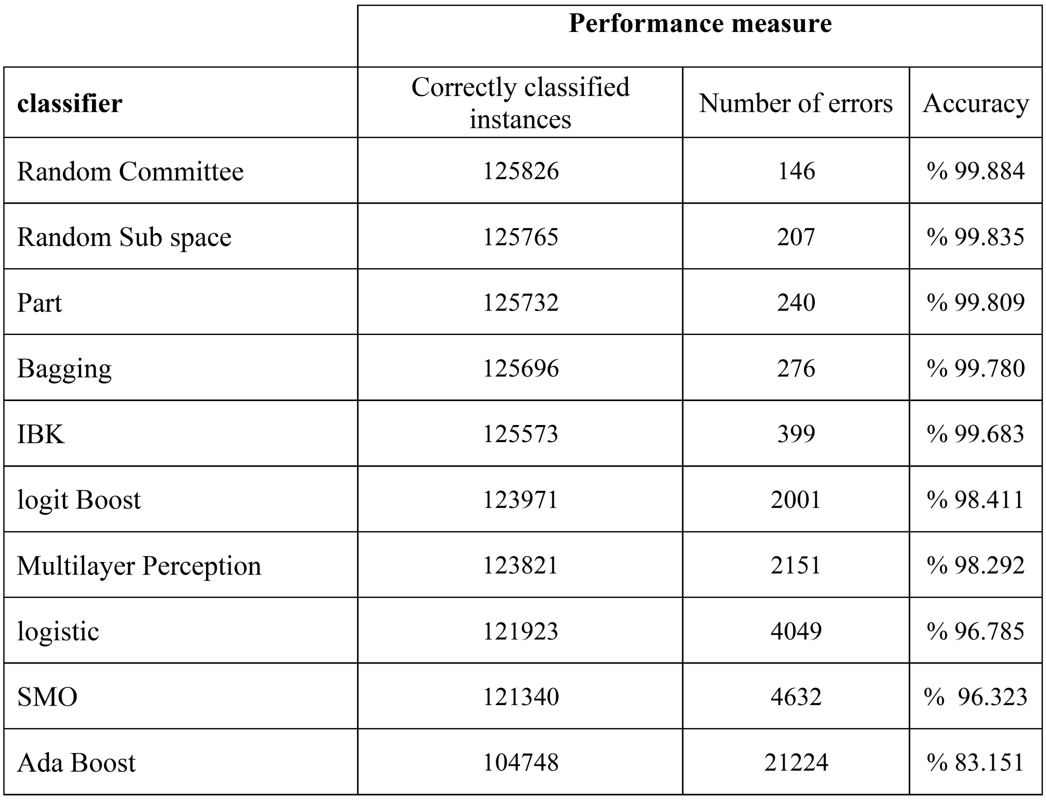

NSL_KDD Dataset
Classification
Test options: 10 fold cross-validation
Modeling the algorithms separately
Selection of better algorithms based on the performance
Ensemble Classifier constructed by voting strategy
Evaluation using different strategies/ selecting the most accurate ensemble classifier
Incorporating cost matrix for intrusion attacks
Counting the errors for each class in the confusion matrix and applying weights in the cost matrix to reduce the error rate
DOS
Prob
R2L
U2R
Selecting the best weights for attack classes
Applying the best 4 weights in cost matrix and then measuring the classification performance

Figure 1: block diagram of the proposed method

Table 3: Performance of the candidate classifiers on NSL-KDD dataset



	Performance measure		
classifier	Correctly classified instances	Number of errors	Accuracy
Random Committee	125826	146	% 99.884
Random Sub space	125765	207	% 99.835
Part	125732	240	% 99.809
Bagging	125696	276	% 99.780
IBK	125573	399	% 99.683
logit Boost	123971	2001	% 98.411
Multilayer Perception	123821	2151	% 98.292
logistic	121923	4049	% 96.785
SMO	121340	4632	% 96.323
Ada Boost	104748	21224	% 83.151

To select proper base classifiers among well-known classifiers, 10-fold cross validation was applied on the classifiers to measure their performance. The accuracy of these candidate classifiers is shown in Table 3 and Figure 2. Among different classifiers, Random Committee achieved the best accuracy of 99.88%. Based on the results in Figure 2, the first 5 classifiers including Random Commitee, Random Subspace, PART, Bagging, and IBK are selected as the base classifiers to construct the ensemble classifier.

After selection of base classifiers, the ensemble classifier is constructed. There are different strategies such as average of probabilities, product of probabilities, maximum probability, minimum probability, and majority vote to fuse the outputs of the base classifiers. Table 4 summarizes the performance of different fusion strategies. As shown in the table, Average probability outperforms than other combining methods. After selecting the fusion strategy, cost sensitive measures such as precision, recall, and F-measure should be should be calculated for each attack, separately. Table 5 shows the cost sensitive measures for the proposed cost sensitive classifier for each class. Based on the simulation, the proposed method achieved a performance near 100% for classification.

4. Conclusions

In this paper, an intrusion detection system is implemented based on the concept of classification in machine learning. Since intrusion detection is naturally a cost sensitive problem, we incorporated a cost matrix into the classification architecture. The elements of this matrix were tuned based on the number of errors the related attacks. Moreover, to improve the generalizability of the classification task, an ensemble classifier was trained instead of a single classifier. Based on 10-fold cross validation, the proposed method achieved a F-measure of 99.9%.

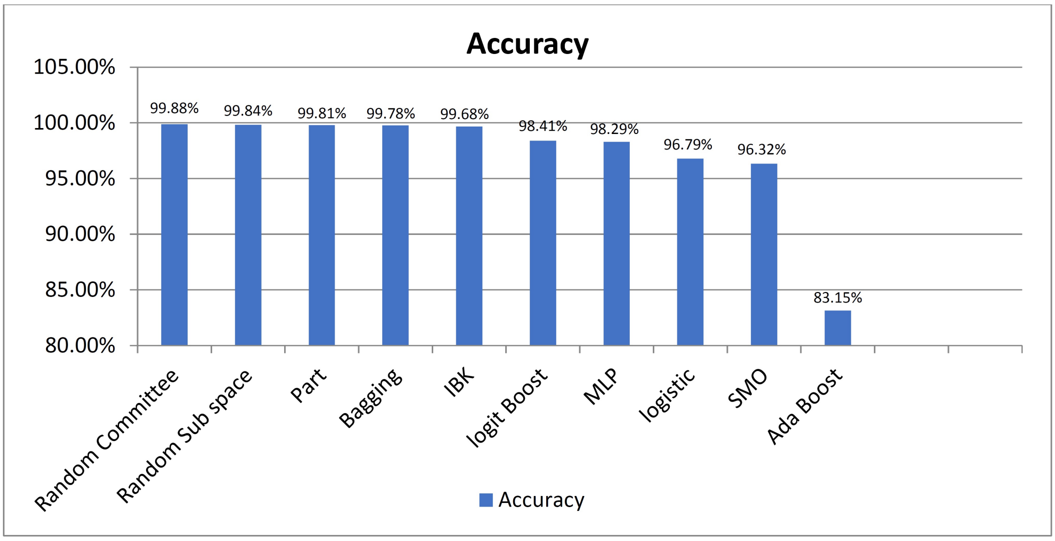


Figure 2: Accuracy of candidate classifiers

Table 4: Comparison of different fusion strategies

Fusion Strategy	Measure		
	Correctly classified instances	Number of errors	Accuracy
Average of probabilities	125844	128	% 99.898
Majority vote	125836	136	% 99.892
Product of probabilities	125755	186	99.827%
Maximum probability	125746	226	% 99.820
Minimum probability	125730	242	% 99.807

Table 5: Cost sensitive measures of the proposed ensemble classifier for each class

Class	Measure		
	Precision	Recall	F-Measure
Normal	0.999	0.999	0.999
DOS	1.000	1.000	1.000
Prob	0.997	0.997	0.997
R2L	0.976	0.981	0.978
U2R	0.731	0.731	0.731
Weighted Average	0.999	0.999	0.999

References

[1] B. Agarwal and N. Mittal, *Hybrid approach for detection of anomaly network traffic using data mining techniques*, Procedia Tech. 6 (2012) 996–1003.

[2] R.G. Bace, *Intrusion Detection*, Sams Paperback Publishing, 2000.

[3] W.–H. Chen, S.–H. Hsu, and H.–P. Shen, *Application of SVM and ANN for intrusion detection*, Comput. Oper. Res. 32(10) (2005) 2617–2634.

[4] J. Hussain and S. Lalmuanawma, *Feature analysis, evaluation and comparisons of classification algorithms based on noisy intrusion dataset*, Procedia Computer Science, 92 (2016) 188–198.

[5] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba and K. Das, *The 1999 DARPA off-line intrusion detection evaluation*, Comput. Networks 34(4) (2000) 579–595.

[6] J. McHugh, *Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory*, ACM Trans. Inf. Syst. Sec. 3(4) (2000) 262–294.

[7] Z. Muda, W. Yassin, M.N. Sulaiman and N.I. Udzir, *A K-means and Naive Bayes learning approach for better intrusion detection*, Inf. Tech. J. 10(3) (2011) 648–655.

[8] A.P. Muniyandi, R. Rajeswari and R. Rajaram, *Network anomaly detection by cascading k-means clustering and* c4.5 *decision tree algorithm*, Procedia Engin. 30 (2011) 174–182.

[9] S.T. Sarasamma, Q.A. Zhu and J. Huff, *Hierarchical Kohonen net for anomaly detection in network security*, IEEE Trans. Syst. Man. Cyber. Part B 35(2) (2005) 302–312.

[10] J. Stanger, and P.T. Lane, *Hack Proofing Linux: A Guide to Open Source Security*, 1^{st} Edition, Elsevier, 2001.

[11] I.S. Thaseen and C.A. Kumar, *Intrusion detection model using fusion of chi-square feature selection and multi class SVM*, J. King Saud Univ. Comput. Inf. Sci. 29(4) (2017) 462–472.