2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)

DDoS Intrusion Detection through Machine Learning Ensemble

Saikat Das*, Ahmed M. Mahfouz, Deepak Venugopal, Sajjan Shiva
Department of Computer Science
The University of Memphis
Memphis, TN, USA 38152
sdas1@memphis.edu, amahfouz@memphis.edu, dvngopal@memphis.edu, sshiva@memphis.edu

***Abstract*— Distributed Denial of Service (DDoS) attacks have been the prominent attacks over the last decade. A Network Intrusion Detection System (NIDS) should seamlessly configure to fight against these attackers' new approaches and patterns of DDoS attack. In this paper, we propose a NIDS which can detect existing as well as new types of DDoS attacks. The key feature of our NIDS is that it combines different classifiers using ensemble models, with the idea that each classifier can target specific aspects/types of intrusions, and in doing so provides a more robust defense mechanism against new intrusions. Further, we perform a detailed analysis of DDoS attacks, and based on this domain-knowledge verify the reduced feature set [27, 28] to significantly improve accuracy. We experiment with and analyze NSL-KDD dataset with reduced feature set and our proposed NIDS can detect 99.1% of DDoS attacks successfully. We compare our results with other existing approaches. Our NIDS approach has the learning capability to keep up with new and emerging DDoS attack patterns.**

Keywords— DDoS, Intrusion Detection System, Network Intrusion Detection System, Ensemble Machine Learning

I. INTRODUCTION

The attackers and the defenders are always fighting each other to exploit and protect the system respectively. Attackers look for system vulnerabilities to exploit the system. On the other end, defenders try to protect the system from that exploitation and suggest a fix. Distributed denial of service (DDoS) has been one of the most prominent attacks, where the perpetrator seeks to make a system, service or resource unavailable from its legitimate users. In DDoS, the system is penetrated in a distributed fashion. The attackers use both traditional and new approaches to accomplish DDoS.

An Intrusion Detection System (IDS) is a program or system which is commonly used for checking and monitoring the target system's activity, and to raise an alarm as soon as it finds any malicious activity. It can be placed both inside or outside of the network perimeter depending on the system defense architecture. Wherever it is placed, the primary goal of the IDS is to detect all types of attacks including DDoS. Over time, attackers change their attack approaches and strategies. To cope up with the attackers' new approaches and to increase the detection accuracy, defenders are implementing new defense techniques, strategies, and methods to build new IDS [1] that can detect the malicious activities.

In this paper, we develop our IDS using Machine learning, which has arguably been the primary driver behind several recent successes in Artificial Intelligence . It is been used in a wide variety of applications including computer vision, natural language understanding, robotics, software engineering, etc. In the security arena, machine learning has been previously used in building IDSs [2, 3]. However, in general, majority of these approaches have focused on learning a single model for intrusions. However, due to the varied nature of intrusions, it may be hard to learn a single model that generalizes to all types. For example, some types of intrusions can be modeled using a simple linear model (e.g. logistic regression) while others may require more complex non-linear models (e.g. support vector machines with kernels). Therefore, our main idea is to train several models that can identify intrusions, and then combine these into a unified system.

The benefits of ensemble learning, i.e., combining multiple classifiers to form a more powerful classifier has been well-studied in the Machine learning community. Dietterich et al. [4] showed that ensembles can perform better than single classifier, and many classification problems have benefited from the idea of combining multiple classifiers. In general, there are two ways to ensemble the classifiers: homogeneous, and heterogeneous. When similar types of classifiers are used to build a training model, it is called homogeneous ensemble (e.g.; bagging, boosting), whereas combining different types of classifiers is called a heterogeneous ensemble (e.g.; stacking). Both homogeneous and heterogeneous ensemble have been used to build IDS. Aburomman et al. [5] mentioned a wide range of ensemble machine learning techniques and methods used to detect network intrusion. However, the key drawback of existing techniques is that they do not use sufficient domain knowledge in conjunction with the ensemble methods. Specifically, several ML methods duffer from the so-called "curse of dimensionality", i.e. as the number of irrelevant features increase, the learned models fail to generalize well. In our context, this means that for newer forms of intrusions (not seen in our training dataset), the models perform poorly. Therefore, we perform a deep analysis of several types of DDoS attacks. Using this, we extract features relevant to these attacks and learn several models using these features. Further, we combine the classifiers in our model using a majority voting method. We empirically show that our proposed approach is much more accurate than existing ML-based intrusion detection methods for the NSL-KDD dataset.

978-1-7281-3925-8/19/$31.00 ©2019 IEEE
DOI 10.1109/QRS-C.2019.00090

Yet, there resides an open research scope to ensemble the classifiers in a better way that can detect the intrusion more accurately. Our contribution in this paper is to build an ensemble model with reduced feature sets [27, 28] in order to increase the DDoS attack detection accuracy and reduce the false positive rate.

The rest of the paper is organized as follows: In Section II, we discuss the state of the art of recent IDS that use ensemble learning and how our contribution is different, and better from their approaches. We propose an IDS framework in Section III and in Section IV, we show our experimentation and observations. We use NSL-KDD [6] dataset and after close observation and verification, we choose reduced feature set [27, 28] that has most relevant DDoS features for this experimentation. TABLE I provides details on these features and how they relate to the DDoS attack. Finally, in Section V, we discuss the pros and cons of our model and conclude the paper with the future research direction.

II. Literature Review

Network Intrusion Detection System (NIDS) detects the abnormal activity of the target system due to the intrusion by an attacker [7]. Signature-based and anomaly-based intrusion detection are two main branches of NIDS. Garcia-Teodoro et al. [8] mentioned different types of anomaly-based intrusion detection and their challenges.

To avoid a human analyst searching through the vast amounts of data in order to find anomalous sequences of network connections, Sinclair et al. [9] built an application that enhanced domain knowledge with machine learning techniques (Genetic algorithms and decision tree) to create rules for an intrusion detection expert system.

Ashraf et al. [10] used machine learning techniques to detect DDoS attacks in software defined network. Suresh et al. [11] compared different types of machine learning algorithms to find a better accuracy in detecting DDoS attack.

In the evolution of NIDS using machine learning, researchers added ensemble machine learning. As mentioned earlier, the two ways of ensembles are homogeneous and heterogeneous. Bagging [12,13] and boosting are used to detect network intrusion where they classified the data with ensemble of similar types of classifiers. But in stacking and hybrid [14] model, ensemble of heterogeneous classifiers are used to train the model. Aburomman et al. [5] provided a detail survey of ensemble-based NIDS including homogeneous, heterogeneous, and hybrid methods and compared those with different types of datasets. To ensemble, different types of classifiers can be used from a variety of classification family, namely: Artificial Neural Network (ANN), Support Vector Machine (SVM) with different kernels, Multilayer Perceptron (MLP), Naïve Bayes (NB), Multivariate Adaptive Regression Splines (MARS), K Nearest Neighbor (KNN), iBK, J48, JRip, PAC, etc. [12, 14, 15, 16, 17].

In addition to choosing the different classifiers to ensemble, reducing features [18, 19] of the dataset can offer better performance in detecting attacks. To detect a DDoS attack, both heterogeneous [5, 15, 18, 20] and homogeneous methods of ensemble have been used. In this paper, we are motivated to create a new ensemble (new combination of classifiers from different classification families) and maintain the minimal number of data features (features that affect DDoS attack) that finally produces lower false positive rate and higher detection accuracy compared to existing results.

III. Proposed Method

The goal of our research is to build an accurate intrusion detection model with a low false positive rate based on an ensemble. Here, we propose an ensemble classifier of four ML classifiers from various classifier families. The selection of these classifiers is based on an our earlier work [21] comparing the performance of different ML algorithms in intrusion detection system domain. The selected classifiers are, MLP (NN), SMO (SVM), IBK (KNN) and J48 (DT-C4.5). In our model, the four classifiers work in parallel and each classifier builds a different model of the data. The outputs of the four predictors are combined by majority voting method to obtain the final output of the ensemble model. Fig. 1. shows the flow of the proposed model which has two main parts: Data Preprocessing and Data Classification process.

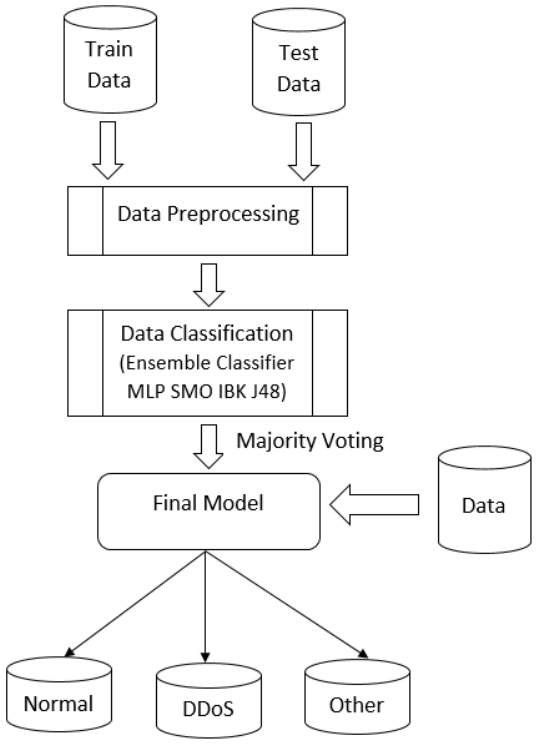

Fig. 1. Ensemble Classifier Model Flow

A. Data Preprocessing

In this section, we briefly discuss the dataset that we used in our experiment, and discuss the data preprocessing by reducing the features from the original dataset to feed the model.

1) Dataset used for experimentation: In this paper, we used NSL-KDD [6] dataset, an improvement of KDD'99 dataset for

TABLE I. SELECTED FEATURES FOR OUR MODEL

No.	Name	Description	DDoS Attack
2	Protocol_type	Protocol used in the connection	Smurf
3	Service	The destination of the network service used	Smurf
4	Flag	The status of the connection	Neptune
5	Src_bytes	Number of bytes transferred from source to destination	Smurf
7	Land	1 if source and destination IP addresses and port numbers are equal and 0 otherwise	Land
8	Wrong_fragment	Total number of wrong fragments in this connection	Teardrop
10	Hot	Number of hot indicators in the content	Back
13	Num_compromised	Number of compromised conditions	Back
23	Count	Number of connections to the same destination host as the current connection in the past two seconds	Smurf
24	Srv_count	Number of connections to the same service (port number) as the current connection in the past two seconds	Smurf
25	Serror_rate	Percentage of the connections which activated the s0, s1,s2 or s3 flags among the aggregated connections in (count)	Neptune
26	Srv_serror_rate	Percentage of the connections that have SYN errors	Neptune
27	Rerror_rate	The percentage of connections that have activated the flag REJ, among the connections aggregated in count	Smurf
28	Srv_rerror_rate	The percentage of connections that have activated the flag REJ, among the connections aggregated in srv_count	Smurf
29	Same_srv_rate	Percentage of the connections to the same service	Neptune
30	Diff_srv_rate	Percentage of the connections which were going to different services, amongst the connections aggregated in (count)	Neptune
33	Dst_host_srv_count	Number of connections having the same port number	Neptune
34	Dst_host_same_srv_rate	The percentage of connections that were to the same service, among the connections aggregated in dst_host_count	Neptune
35	Dst_host_diff_srv_rate	The percentage of connections that were to different services, among the connections aggregated in dst_host_count	Neptune
36	Dst_host_same_src_port_rate	The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_c ount	Smurf
38	Dst_host_serror_rate	The percentage of connections that have activated the flag s0, s1, s2 or s3, among the connections aggregated in dst_host_count	Neptune
39	Dst_host_srv_serror_rate	Percentage of the connections which have activated the s0, s1, s2 or s3 flags among the connections aggregated in (dst_host_srv_count)	Neptune
40	Dst_host_rerror_rate	The percentage of connections that have activated the flag REJ, among the connections aggregated in dst_host_count	Smurf
41	Dst_host_srv_rerror_rate	The percentage of connections that have activated the flag REJ, among the connections aggregated in dst_host_srv_c ount	Smurf

our experimentation. The dataset contains hundreds of thousands of connection records. Each of the single that defines either a normal or anomalous state. A detailed description of the features is provided in [22]. There are three types of feature sets: Basic features of individual TCP connections, Content features within a connection suggested by domain knowledge, and Traffic features computed using a two-second time window.

For the data classification process, we need three types of datasets: training dataset, test dataset, and a validation dataset to validate the classifier which is an optional dataset. All training, and testing datasets we found from [6] contain seven files are listed below:

- ✓ KDDTrain+.ARFF: The full NSL-KDD train set with binary labels in ARFF format
- ✓ KDDTrain+.TXT: The full NSL-KDD train set including attack-type labels and difficulty level in CSV format
- ✓ KDDTrain+_20Percent.ARFF: A 20% subset of the KDDTrain+.arff file
- ✓ KDDTrain+_20Percent.TXT: A 20% subset of the KDDTrain+.txt file
- ✓ KDDTest+.ARFF: The full NSL-KDD test set with binary labels in ARFF format
- ✓ KDDTest+.TXT: The full NSL-KDD test set including attack-type labels and difficulty level in CSV format

- ✓ KDDTest-21.ARFF: A subset of the KDDTest+.arff file which does not include records with difficulty level of 21 out of 21
- ✓ KDDTest-21.TXT: A subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21

2) Dataset pre-processing: Most machine learning models work with numeric values only. In the data preprocessing phase, we have converted the feature values from nonnumeric to numeric. As the focus of this research is to implement a DDoS attack detection model, we analyzed and verified the most relevant DDoS features [27, 28] from the NSL-KDD dataset. In the dataset, data-instances that contain the class labels Back, Land, Neptune, Smurf and Teardrop correspond to DDoS. TABLE II shows the most relevant [27] features under each class label for DDoS atack. After comprehending all class labels, the full DDoS feature set is 2, 3, 4, 5, 7, 8, 10, 13, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 38, 39, 40, and 41.

TABLE II. RELEVANT FEATURES FOR DDOS ATTACKS [27]

Class Label	Most Relevant Features
Land	7
Smurf	2, 3, 5, 23, 24, 27, 28, 36, 40, 41
Neptune	4, 25, 26, 29, 30, 33, 34, 35, 38, 39
Teardrop	8
Back	10, 13
Full Set	2, 3, 4, 5, 7, 8, 10, 13, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 38, 39, 40, 41

B. Data Classification

In the data classification section, our model classifies the data one by one with individual classifiers, and then classifies with ensemble classifiers where the classifiers are combined with majority voting method.

1) Data classification with individual classifier: To begin with data classification, we classified our dataset with individual classifiers. There are several models exist in machine learning but in this paper, we use four different classifiers from various classifier families: MLP (NN), SMO (SVM), IBK (KNN) and J48 (DT-C4.5). We tuned the performance accuracy with at least ten classifiers from different classifier families and choosed best four.

- *C4.5 (J48 in Weka):* C4.5 is an algorithm that is used to generate a decision tree. By using a set of training data, it builds a decision tree like ID3 (Iterative Dichotomiser 3) [23] but it has the number of improvements over ID3: handle both continuous and discrete attributes, handle training data with missing attribute values, handle attribute with differing costs, prune trees after creation. In Weka [24], J48 classifier is implemented to build a decision tree by using C4.5 algorithm.
- *KNN (iBK in Weka):* KNN stands for k-nearest neighbors which is mostly used in pattern recognition. KNN is a non-parametric method that is used for both classification and regression purposes.
- *NN (MLP in Weka):* NN stands for Neyral Network which is a set of algorithms that is modeled after the human brain and deigned to recognize the pattern. MLP is the Weka implementation for NN.
- *SVM (SMO in Weka):* Suport Vector Machine (SVM) is an well known classifier that is used in supervised learning data classification. In Weka, SMO is used for SVM classification. SVM is a discriminative classifier that is defined by a hyperplane. For a given labeled dataset, this classifier provides an optimal hyperplane as output. There are different types of kernels that are used in SVM. We used poly-kernel in our implementation.

C. Ensemble with majority voting

As mentioned earlier, the ensemble model consists of four different classifiers that work in parallel. Each classifier builds a different model of the data based on the preprocessed train dataset. To build the models, each classifier was tested using the 10-folds cross-validation technique within the dataset, where the dataset gets divided into 10 folds or subsets. Any 9 subsets were used as training sets and the remaining subset was used as the test set. More specifically, each fold was analyzed, and the total score results determined the average performance out of 10-folds.

Majority voting is one of the traditional and common way to combine the classifier. In our experimentation, the outputs of the four predictors were combined and we used WEKA majority voting to obtain the final output of the ensemble model.

IV. EXPERIMENTAL RESULTS

The proposed model has been implemented using WEKA [20], a data mining tool which was running on a PC with Intel(R) CORE(TM) i5-6600K CPU @ 3.50GHz, 3.50 GHz, 8 GB RAM installed and running a 64-bit Windows 10 OS, x64-based processor.

TABLE III. NO OF SAMPLES FOR NORMAL AND DOS CLASSES

Class	Training Set	Occurrence Percentage	Testing Set	Occurrence Percentage
Normal	67343	53.46 %	9711	43.08%
DDoS	45927	36.46 %	7460	33.08%
Other	12703	10.08 %	5373	23.85%
Total	125973	100%	22544	100.0%

The classifiers were trained on the training dataset provided by NSL-KDD using Stratified Cross-Validation of 10-folds and the produced models were tested on the testing dataset that is also provided by NSL-KDD. TABLE III. shows the number of attack records associated with each class in both train and test datasets.

In terms of performance evaluation, Confusion Matrix, True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR), Precision, Recall, F-Measure and Receiver Operating Characteristics (ROC) curve are used very frequently.

Confusion Matrix has three main terms: Sensitivity, Specificity, and Accuracy which can be defined as follows:

$$Sensitivity = \frac{TPR}{(TPR+FNR)} \quad (1)$$

$$Specificity = \frac{TNR}{(TNR+FPR)} \quad (2)$$

$$Accuracy = \frac{(TPR+TNR)}{(TPR+TNR+FPR+FNR)} \quad (3)$$

Also, Precision, Recall and F-Measure are another three important performance metrics that are used to evaluate a model. Those terms can also be defined by TP, TN, FP, FN from equations (4), (5) and (6).

$$Precision\ (P) = \frac{TP}{(TP+FP)} \quad (4)$$

$$Recall\ (R) = \frac{TP}{(TP+FN)} \quad (5)$$

$$F - Score = \frac{2PR}{PR} \quad (6)$$

ROC curve is a well-known evaluation measure that visualizes the relation between True Positive (TPR) and False Positive (FPR) rates.

In our experimentation, we used those performance metrics to evaluate our model and compared it to existing researches. First, we show the comparison of individual classifier's (MLP, SMO, IBK, J48) performance in terms of Accuracy, TPR, FPR, Precision, Recall, F-Measure, ROC area in TABLE IV. From this table we see that when single classifier was applied to build the model, J48(decision tree) classifier had better performance compared to other classifiers.

TABLE IV. PERFORMANCE METRICS WHEN CLASSIFIER USED

Classifier	**MLP**	**SMO**	**IBK**	**J48**
Accuracy	96.5%	95.73%	97.83%	97.89%
TPR	0.973	0.966	0.979	0.979
FPR	0.051	0.060	0.022	0.022
Precision	0.965	0.960	0.979	0.979
Recall	0.973	0.966	0.979	0.979
F-Measure	0.969	00963	0.979	0.979
ROC Area	0.973	0.953	0.978	0.979

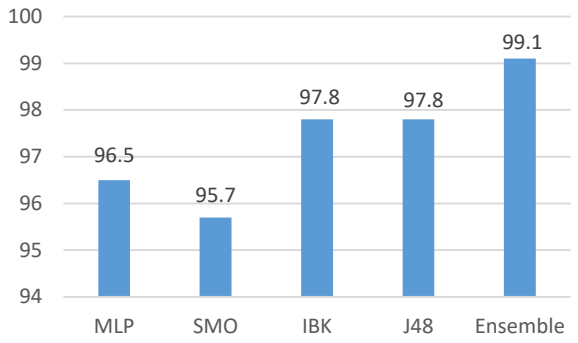

Fig. 2. Single and Ensemble Classifiers' Accuracy

From Fig. 2., we see that ensemble classifier outperformed with respect to individual classification's accuracy.

The experimental results show that the proposed ensemble IDS model was able to correctly classify 125679 instances which is 99.77 % of the data. The number of incorrectly classified instances was 294 that is 0.23 % of the data. Fig. 2. and TABLE IV show that the ensemble model outperforms every single classifier used in terms of Accuracy, TPR, FPR, Precision, Recall, F-Measure, and ROC curve.

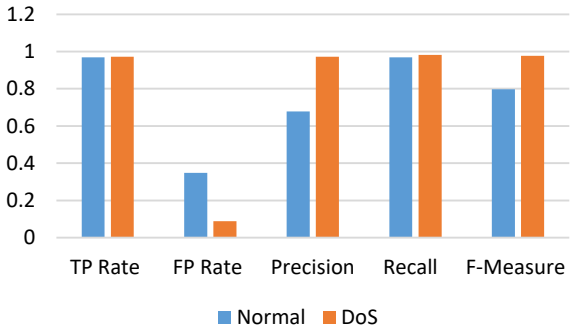

Fig. 3. Comparison of Normal vs DoS Performance Metrics for Ensemble Classifier

In Fig. 3., we present some other performance metrics for the proposed model and in Fig. 4. and Fig. 5., we show the ROC curve for detecting both DoS and Normal classes.

From Figues 4 and 5 it is clear that the ensemble classification model is much better for DDoS intrusion detection in terms of accuracy, TPR, and FPR than any single classifier. Indeed, our model has the better detection accuracy compared to other models [5, 10, 15, 20]. For instance, we increased the model's accuracy rate compared to [11] from 98.7% to 99.1%, and lessen the false positive rate mentioned in [18] from 0.42% to 0.088%.

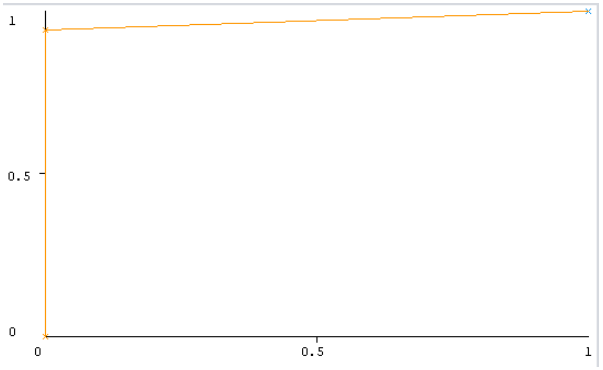

Fig. 4. ROC Curve for DDoS classification

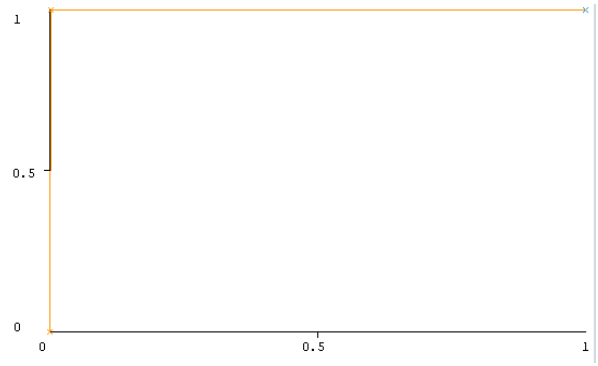

Fig. 5. ROC Curve for Normal Classification

V. Conclusion

Machine learning, a subset of artificial intelligence takes the intrusion detection study to a new dimension. Ensemble of machine learning classifiers including reduced feature set often produce a better detection accuracy rate compared to single classifier. In detecting DDoS attack, machine learning based IDS has promising outcomes. In this paper, we proposed a NIDS, which has the capability to detect a DDoS attack by using ensemble classifiers and a reduced feature dataset. In our experimentation, we used NSL-KDD dataset with a reduced feature set [27, 28] to detect only DDoS attacks. Based on our domain knowledge, we used the most relevant features [27] that can only affect a DDoS attack. To create a variety of classifiers, we chose MLP, SMO, IBK, J48, IBK which are from various classification family. Finally, 10 -folds cross validation along with majority voting helps to combine those classifiers. We compared with existing results and found that our proposed model has better detection accuracy with a lower false positive rate. We plan to expand our experimentation to detect multiple types of attacks with various types of datasets [25] including online data and then we will use each of the detectors as an agent [26] to monitor the system behavior that will eventually create a distributed IDS.

References

[1] Axelsson, Stefan. Intrusion detection systems: A survey and taxonomy. Vol. 99. Technical report, 2000.

[2] Eskin, Eleazar, et al. "A geometric framework for unsupervised anomaly detection." Applications of data mining in computer security. Springer, Boston, MA, 2002. 77-101

[3] Eskin, Eleazar. "Anomaly detection over noisy data using learned probability distributions." (2000).

[4] Dietterich, Thomas G. "Ensemble methods in machine learning." International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000

[5] Aburomman, Abdulla Amin, and Mamun Bin Ibne Reaz. "A survey of intrusion detection systems based on ensemble and hybrid classifiers." Computers & Security 65 (2017): 135-152.

[6] NSL-KDD dataset [online] available: http://www.unb.ca/cic/datasets/nsl.html Accessed on 10/21/2018.

[7] Mukherjee, Biswanath, L. Todd Heberlein, and Karl N. Levitt. "Network intrusion detection." IEEE network 8.3 (1994): 26-41.

[8] Garcia-Teodoro, Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security 28.1-2 (2009): 18-28.

[9] Sinclair, Chris, Lyn Pierce, and Sara Matzner. "An application of machine learning to network intrusion detection." Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99). IEEE, 1999.

[10] Ashraf, Javed, and Seemab Latif. "Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques." 2014 National Software Engineering Conference. IEEE, 2014.

[11] Suresh, Manjula, and R. Anitha. "Evaluating machine learning algorithms for detecting DDoS attacks." International Conference on Network Security and Applications. Springer, Berlin, Heidelberg, 2011.

[12] Syarif, Iwan, et al. "Application of bagging, boosting and stacking to intrusion detection." International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, Berlin, Heidelberg, 2012.

[13] Gaikwad, D. P., and Ravindra C. Thool. "Intrusion detection system using bagging ensemble method of machine learning." 2015 International Conference on Computing Communication Control and Automation. IEEE, 2015.

[14] Borji, Ali. "Combining heterogeneous classifiers for network intrusion detection." Annual Asian Computing Science Conference. Springer, Berlin, Heidelberg, 2007.

[15] Chan, A. P. F., et al. "Comparison of different fusion approaches for network intrusion detection using ensemble of RBFNN." Proceedings of 2005 international conference on machine learning and cybernetics. Vol. 6. 2005.

[16] Haq, Nutan Farah, Abdur Rahman Onik, and Faisal Muhammad Shah. "An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA)." 2015 SAI Intelligent Systems Conference (IntelliSys). IEEE, 2015.

[17] Mukkamala, Srinivas, Andrew H. Sung, and Ajith Abraham. "Intrusion detection using an ensemble of intelligent paradigms." Journal of network and computer applications 28.2 (2005): 167-182.

[18] Osanaiye, Opeyemi, et al. "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing." EURASIP Journal on Wireless Communications and Networking 2016.1 (2016): 130.

[19] Zhang, Fengli, and Dan Wang. "An effective feature selection approach for network intrusion detection." 2013 IEEE Eighth International Conference on Networking, Architecture and Storage. IEEE, 2013.

[20] Jia, Bin, et al. "A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning." Journal of Electrical and Computer Engineering 2017 (2017).

[21] [Accepted] Ahmed M. Mahfouz, Deepak Venugopal, Sajjan G. Shiva. "Comparative Analysis of ML Classifiers for Network Intrusion Detection" ICICT 2019: Fourth International Congress on Information and Communication Technology.

[22] Lin, Shih-Wei, et al. "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection." Applied Soft Computing 12.10 (2012): 3285-3290.

[23] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[24] Holmes, Geoffrey, Andrew Donkin, and Ian H. Witten. "Weka: A machine learning workbench." (1994).

[25] Vanerio, Juan, and Pedro Casas. "Ensemble-learning approaches for network security and anomaly detection." Proceedings of the Workshop

on Big Data Analytics and Machine Learning for Data Communication Networks. ACM, 2017

[26] Das, Saikat, and Sajjan Shiva. "CoRuM: Collaborative Runtime Monitor Framework for Application Security." 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). IEEE, 2018.

[27] Noureldien, N. A., and I. M. Yousif. "Accuracy of machine learning algorithms in detecting DoS attacks types." Science and Technology 6.4 (2016): 89-92.

[28] Olusola, Adetunmbi A., Adeola S. Oladele, and Daramola O. Abosede. "Analysis of KDD'99 intrusion detection dataset for selection of relevance features." Proceedings of the World Congress on Engineering and Computer Science. Vol. 1. WCECS, 2010.