Mining network traffics for intrusion detection based on Bagging ensemble Multilayer perceptron with Genetic algorithm optimization

Mehdi Moukhafi[†], Khalid El Yassini[†] and Seddik Bri[††]

[†]Informatics and Applications Laboratory (IA), Department of Mathematics and Computer Science, Faculty of Sciences, Moulay Ismail University, Meknes, Morocco
[††]Materials and Instrumentations (MIN), Department of Electrical Engineering, Superior School of Technology: ESTM, Moulay Ismail University, Meknes, Morocco

Summary
Due to the frequency of malicious network activities and network policy violations, Intrusion Detection Systems (IDS) have become a necessity in computer security systems because of the increase in unauthorized accesses and attacks. Current IDSs are mainly based on techniques built on heuristic rules called signatures to detect intrusions in a network environment. These approaches based signature could only detect known attacks and referenced above. Since there is no signature for new attacks, other approaches must be taken in consideration, such as algorithms learning machine to detect a known and unknown attacks. Recent researches suggest combining multiple classifiers could have a better performance. In this paper, we propose a method of intrusion detection based on a combination of GA(Genetic algorithm) and Bagging ensemble MLP (Multilayer Perceptron) Neural Network to develop a model for intrusion detection system. The performance of the proposed method of intrusion detection was evaluated on all NSL-KDD test set and NSL-KDD train set is used for training the GA-Bagging-MLP model. Additionally, the performance of this approach has been analysed and compared with a number of existing approaches tested in the same data set. The results show that proposed method outperforms the existing approaches.

Key words:
Machine learning Based Intrusion Detection; Parameters optimization; Genetic algorithm; Multilayer perceptron Neural Network

1. Introduction

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1]. The number of intrusions into computer systems is constantly growing; the attacks carried out by malicious users to exploit the vulnerabilities of these systems are more and more frequent. In this context, intrusion detection systems (IDS) have emerged as a group of methods that combats the unauthorized use of a network's resources. Recent advances in information technology, especially in machine learning, have produced a wide variety of machine learning methods, which can be integrated into IDS.

Intrusion detection system (IDS), presented for the first time by Anderson in 1980 [1], and later formalized by Denning [2], is hardware or software system that collects information from computer network analyses it and finds whether there exists any abnormal activity, which violates the integrity, availability and confidentiality of the information system. It is a network security mechanism for detecting, preventing, and repelling unauthorized access to a communication or computer network. The IDS play a crucial role in maintaining a safe and secure network. It aims to identify and respond to malicious activity that compromises network and computer security [1] and report security violations to a system administrator.

The main goal of IDS is to alarm the network administrator if any suspicious activity happening. It is also bound to provide the following services:

- Audit the system and vulnerabilities;
- Evaluating the integrity of network;
- Tracking anomalies;
- Observing and analysing network and system activities;
- Providing user friendly interface for security management.

There are two types of IDSs: Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host.

The major drawback of Actual IDS's, based on heuristic rules (signature based system), that cannot detect novel attacks whose signature is not available in the rule base [3]. To overcome the mentioned problem above, many techniques have been developed, especially using Artificial Intelligence (AI) technologies.

In recent decades, the intrusion detection has benefited from the idea of mining the network traffic with mechanisms capable of learning artificially to detect attacks and even their variants. Instead of trying to detect

Manuscript received May 5, 2018
Manuscript revised May 20, 2018

the intrusion by comparing its signature with a database, it seeks to establish a model of behavior of each type intrusion. In other words, to detect the attacks, it is necessary to study the behavior of this last one and the behavior of the legitimate traffic in order to establish a model capable of classifying the traffic circulating within the network in two categories: normal and abnormal.

This paper presents intrusion detection system model based on neural network, using MLP (Multilayer Perceptron) optimized with Genetic Algorithm. MLP classifier is employed to classify network traffic into normal and abnormal connections. GA is used to optimize the architecture of MLP classification model.

The paper is organized as follows. Related works are discussed in Section 1. Section 2 gives an overview of methods used in this work and a description of the proposed method. Section 3 describes the system model. Section 4 evaluates the proposed system and at last. Section 5 presents the conclusion of this work.

2. Review of literature

Anomaly detection has been an important subject in intrusion detection research. Various anomaly detection approaches based on Artificial Intelligence technologies are utilized in IDSs due to their flexibility and learning capability. These intelligent systems construct a general model of existing patterns which will be able to detect new attacks. Additionally, the IDS's performances are considerably improved at the network level. The machine learning algorithm obtains a good detection performance in terms of classifying the flow of a network into normal or abnormal behaviors.

Folino et al. [1] introduced a distributed data mining algorithm to ameliorate detection accuracy when classifying normal or abnormal network activities. This algorithm is based on a genetic programming (GP) extended with ensemble paradigms. GP is used to build a network profile by combining different classifiers that together provide complementary information.

Monowar et al. [2] introduced a procedure based on a mutual and general information, an entropy function selection technique for selecting a non-redundant subset of features, based on a clustering and trees to generate a set of reference points and a function of aberrant score to classify incoming network traffic to identify anomalies.

Govindarajan and Chandrasekaran [3] proposed a hybrid system, that combined a prediction of various classifiers. To manufacture their IDS they implemented three modules, the first, based on the RBF neural network, in charge of detection of U2R attacks, the second one is based on SVM for DOS attacks and the third is an RBF-SVM hybrid to recognize a normal traffics while the latest module is an ensemble classifier, RBFSVM, that detects a Probe and R2L attacks. In addition, the authors implemented a best-first search (BFS), for feature selection.

Kuang et al. [4] proposed a solution based on a combination of the SVM model with kernel principal component analysis (KPCA) and genetic algorithm. KPCA was used to reduce the dimensions of feature vectors, whereas GA was employed to optimize the SVM parameters.

Pervez and Farid [5] presented a hybrid approach based on feature selection and classification for multiple classes NSL-KDD, using support vector machines in a one against rest multi-class configuration (OAR-SVM). The feature selection was implemented following the LOO method.

Nadiammai and Hemalatha [6] established a DOS attack detection with a hybrid mechanism which is based on two steps, the first is the submission of the traffic flow which has an intrusion detection system based signature (SNORT) for preliminary detection of the incoming flow and detecting known attacks, the second step is the recovery of the classified flows as legitimate and applies a classification based supervised learning (SVM) with the use of the kernel function Radial basis function.

Kanakarajan and Muniasamy [7] have applied a novel tree ensemble technique called GAR-Forest for both binary and multi-class classification intrusion detections. It is a solution based on on a greedy randomized adaptive search procedure with annealed randomness classifier. The GRASP was deployed to generate a set of randomized adaptive decision trees.

Gaikwad and Thool [8] implemented homogenous ensemble classifier to solve the intrusion detection problem. The bagging classifier is constructed with partial decision tree classifier and the Genetic Algorithm is used to select the relevant features from the NSL-KDD99 dataset.

We can mention Intrusion Detection Technique used by Saied et al. [9] to detect DDOS attacks known and unknown in real time, based on the use of an artificial neural network (ANN) and specific characteristics (models) that differ between legitimate traffic and DDOS requests using learning by back propagation coupled with a sigmoid activation function. The authors have selected three ANN topological structures, one for the most used protocols (TCP, UDP, ICMP) in DDOS attacks, each one with three layers (input, hidden and output). The number of nodes in each topological structure is different. The ICMP topological structure consists of three inputs and four hidden nodes, the topology structure TCP consists of five inputs and four hidden nodes and the topological structure of UDP consists of four inputs and three hidden nodes that treat the calculation process with respect to input and output nodes. The output layer consists output node for the attacks and an output node for legitimate traffic.

Wathiq et al. [10]proposed a solution based on hybrid SVM and Extreme Learning Machine model learned with data set built by a modified K-means which is used to build new small training datasets representing the entire original training dataset.

Aygun and Yavuz [11] proposed two deep learning based anomaly detection models using autoencoder and denoising autoencoder respectively. The key factor that directly affects the accuracy of the proposed models is the threshold value which was determined using a stochastic. The AE was trained with only normal data to produce binary ids (normal / abnormal).

Rashid et al. [12] have proposed a method for Network Intrusion Detection System (NIDS) based on the concept of DNA sequencing. Firstly, they converted the network traffic data into a form of DNA sequence using Cryptography Encoding Method. Secondly, Teiresias algorithm is used to detect sequence for each network traffic class, finally, Horspool matching algorithm is utilized to classify the data into attack or normal.

The present study proposes a new hybrid ensemble approach, GA-Bagging-MLP, which is based on two strategies, genetic optimization and bagging and multilayer perceptron (MLP) as base learner. As there are two steps, a procedure to find the most important parameters of MLP and bootstrap selection of instances encouraging diversity in MLP ensemble classifier, the base learner's combination strategy for bagging is majority vote. The presented method is evaluated by the NSL-KDD, to show that the system performance intrusion detection is significantly promoted by our method. Compared with other methods based on the same dataset, the proposed model shows high detection rate and its accuracy.

3. Proposed work

In this section, we first describe Multilayer Perceptron algorithm, Genetic algorithm, Bagging ensemble learned with NSL-KDD data set and then we discuss our proposed method for IDS.

3.1 Methodology

- *Multilayer Perceptron*

Artificial Neural Network (ANN) [13] is defined as an information processing system that has characteristics resembling biological nervous systems. It is composed by a set of simple computational units that are highly interconnected. The typical ANN's information-processing unit (neurons) is organized in three groups, or layers: input, hidden, and output. The ANN provides a new technology to help solve problems that require thinking of experts and computer based routine.

Multilayer Perceptron (MLP) [14] is the neural network used in this work. It is a kind of feed forward neural network. MLP is the most popular type of ANN, because of its simple construction and reduced training requirements and mechanism. It is composed by a set of simple computational units that are highly interconnected, MLP is a layered feed forward network typically trained with static back propagation (BP). Such networks have found their way into countless applications requiring static pattern classification. The MLP model is a flexible type of ANN composed of one input layer, one or more hidden layers, and one output layer. Figure 1 shows the main components, the hidden units receive a weighted sum of the inputs and apply an activation function to it. Then, the output units receive a weighted sum of the hidden units output and apply an activation function to this sum. Figure 1 depicts the architecture of the MLP neural network model.

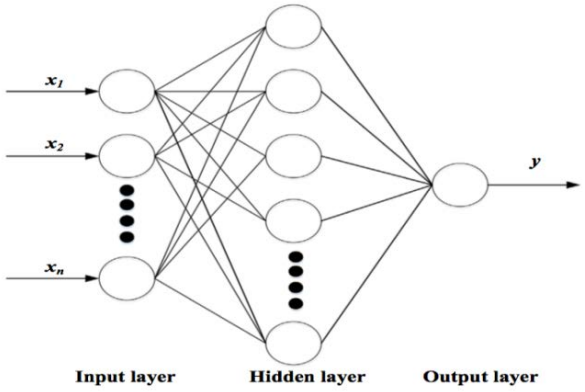


Fig. 1 Multilayer Perceptron neural-network architecture

Each layer has a number of neurons and each neuron is fully interconnected with weighted connections to the neuron in the subsequent layer. Therefore, a general expression of the network can be given as:

$$Y = F\left(\sum_{j=1}^{m} W_{jk}.F\left(\sum_{i=1}^{n} W_{ij}X_i + B_j\right) + B_k\right) \qquad (1)$$

Where Wjk are the weights between hidden and output layer, Wij are weights between input and hidden layers; and Xi are input variables; m is the number of neurons in a hidden layer; n is the number of neurons in an input layer; k is the number of output classes, Bj and Bk are the bias values of the neurons in the hidden and output layers, respectively; F is the transfer function and Y is the output.

- *Genetic algorithm*

Genetic algorithm (GA) is a directed heuristic search technique, it is a general adaptive optimization search

methodology based on a direct analogy to Darwinian's principle of evolution of fittest to optimize a population of candidate solutions towards a predefined fitness [15]. It is an optimization technique that attempts to replicate natural evolution processes in which the chromosome with the considered best characteristics to adapt to the constraints are more apt to reproduce and survive. The main theme behind this technique is to find an optimal solution for complex problems.

The algorithm of evolution by natural selection can be formulated in many ways, but the following encapsulates its essential elements:

- Successive generations can differ from previous ones. Children inherit characteristics of their parents. But combining and mutating these characteristics introduce variation from generation to generation.

Less fit individuals are selectively eliminated ('survival of the fittest').

The procedure for a genetic algorithm can be given as:

Initialization - Create an initial population. This population is usually randomly generated of n chromosomes (suitable solutions for the problem)

Repeat:

•Selection - Select two parent chromosomes from a population, based on their fitness (the better fitness, the bigger chance to be selected)

•Crossover - create new individuals by combining aspects of the selected parents.

•Mutation - The algorithm creates mutation children by randomly changing the genes of individual parents. Mutation typically works by making very small changes at random to an individual genome.

•Evaluation - Each chromosome x of the population is then evaluated and calculate a 'fitness'.

Test - If the end condition is satisfied, stop, and return the best solution in current population, else return to selection step.

- *Bagging*

Bagging was created by Breiman [16] and is an acronym for bootstrap aggregating, is a ensemble meta-algorithm used to improve an accuracy of machine learning algorithms. This meta-algorithm consists of creating bootstrap replicas of the training set and then training a different classifier with each replica. Fusing of particular classifiers is achieved by the use of a majority vote on their selections. Thus, for any example input, the ensemble's decision is the class selected by the greatest number of classifiers. The bootstrap sets are built randomly from the original training set using substitution.

- *Benchmark NSL-KDD*

The NSL-KDD intrusion data set [17], which was provided to solve some shortcomings in KDD'99 [22], particularly that it's training and test sets contained a huge number of redundant records with about 78% and 75% of the records being duplicated in the training and test sets, respectively. That was summarized into record connections where each connection is a single row vector consisting of 41 features and one class attribute as normal or attack or the attack types. The attack classes presented in this data set are grouped into four categories, namely, denial of service attack, probes, remote to user attack and user to root attack. In this research, NSL-KDD intrusion dataset is used for GA-Bagging MLP classifier based Intrusion Detection System. The training is performed on KDDTrain+ data which contain 22 attack types and testing is performed on KDDTest+ data which contains additional 17 attack type (table 1).

Table 2: List of intrusions in training and testing data

Intrusions existing in both training and testing data	**Intrusions existing only in testing data**
back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster	apache2, httptunnel, mailbomb, mscan, named, perl, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop, xterm

These attacks can be categories in four different types with some common properties:

Denial of Service Attack (DoS): it is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

User to Root Attack (U2R): it is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

Remote to Local Attack (R2L): it occurs when an attacker with the ability to send packets to a machine over a network but without an account on that machine exploits some vulnerability to gain local access as a user of that machine.

Probing Attack: it is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

3.2. Proposed approach

Multilayer Perceptron is using only numerical data remaining in the same range. For this reason, the proposed method creates a pre-processing module to transform value of features of each packet from characters to numeric value. After that, a Normalization process is performed on the numeric values to make it in the same range.

Initially in the NSL-KDD, the features 2,3 and 4 (protocol type, Service, Flag) correspond to chains of characters

(String). They are converted to numerical data's. The correspondence is specified by Table 2.

Table 2: replacing the String in features 2, 3 & 4

String	Numeric replacement	String	Numeric replacement
Protocol Type			
TCP	1	**ICMP**	3
UDP	2	-	-
Services			
HTTP	1	**Finger**	5
Domain_u	2	**Time**	6
Private	3	**Domain**	7
Ftp_data	4	**ftp**	8
Flag			
SF	1	**S0**	3
Rej	2	**S1**	4

In our approach, standardization was performed by scaling numerical characteristics relative to their mean and standard deviation; this prevents features with large numeric values to dominate other features. The Feature scaling method is used to normalize numeric values to range between MinX and MaxX that are the minimum and maximum values for feature X. We first convert [MinX, MaxX] to a new range [New MinX New MaxX], according to Eq. (2) where each value of x in the original range is converted to a new value [18].

$$newx = \frac{x - MinX}{MaxX - MinX} \qquad (2)$$

Genetic algorithms (GA) can be used as a powerful optimization tools to a complex problem with each other in order to achieve increasingly better results. In the present work, the GA is used to look for the best architecture of the MLP classification model. Indeed, each chromosome represents a solution of the number of neurons in the input layer (selection of optimal features) in formula (1) and it is represented by n and the number of hidden layers.

Firstly, the algorithm creates randomly a population; each individual is an M-dimensional vector which represents a candidate's solution. The Multilayer Perceptron builds a wide range candidates solution to evaluate its performance until the optimal solution. The GA guides the selection of potential parameters that lead to better prediction accuracy. The algorithm uses the best individuals to contribute to the generation of the next generation. Thus, on average, each successive population of candidate particles is better than its predecessor. Algorithm 1 describes the first step of proposed solution.

Algorithm 1: Multilayer Perceptron Optimized by Genetic Algorithm

Input: Training data set (KDDTrain+)
Output: Best parameters

InitPopulation (Pop): Initializes the population;
EvalPopulation (Pop): Evaluates the fitness of each individual in the population;

1: ***While*** not termination ***do***

`//Elitism based selection`
2: ***Select*** best-ranking individuals **from Pop and delete** others;

`//Crossover`
3: ***for*** i=1 **to** (Pop.Length)/2 ***do***
4: ***Randomly select*** two solutions Xa and Xb from population;
5: ***Generate*** Xc and Xd by crossover to Xa and Xb;
6: ***Save*** Xc and Xd in Pop;
7: ***endfor***

`//Mutation`
8: ***for*** i=1 to Pop.Length ***do***
9: ***Select*** Solution Xi from Pop;
10: ***Mutate*** a set of bits randomly and generate Xi';
11: ***Update*** Xi by Xi' in Pop;
12: ***endfor***

`//Evaluation`
13: ***for*** i=1 to Pop. Length **do**
14: ***Train*** MLP with using solution Xi from Pop;
15: ***Evaluate*** MLP model and calculate accuracy;
16: ***endfor***
endwhile

The classifier is trained with an individual and evaluated by his fitness; the evaluation function used is the maximization of the accuracy. Each individual is given a probability of being selected that is directly proportionate to its fitness score. After this step, the algorithm selects half of the individuals evaluated for applying a crossover to produce offspring with them.

After selection and crossover, the algorithm applies a mutation to the reproduced population by randomly changing 1 percent of genes of each individual in order to ensure that the individuals are different from the previous population. This process continues until the MLP performance is satisfactory.

The first step serves only as a procedure for extracting optimal parameters for the multilayer perceptron, since the model that we have developed up to now suffers from high classification inaccuracy. For that we have elaborated a

model based on homogeneous multi-classifier, Algorithm 2 describes the second step of proposed solution.

Algorithm 2: Bagging ensemble Multilayer Perceptron classifier

Training data set: KDDTrain+
Testing data set: KDDTest+

Input: Multilayer Perceptron classifier L;
Number of bootstrap T;
Best MLP parameters P;
Training data set S;
Training data set S;
Labels $\mathbf{Y}$;

Output: classifier L^*

1: *for* i=1 to **T**

2: S_i = Bootstrap (**S**) % generate bootstrap from sample **S**
3: S_i^k=RS(S_i,k) % Random generate a subspace from S_i
4: L_i = Train a classifier on S_i^k via L with best parameters P

5: *endfor*

$$L^* = argmax \sum_{i:Li(x)=y \atop y \in Y} 1$$

The Bagging algorithm is a kind of general ensemble learning algorithm, whose structure can be defined as follows. First, it randomly samples the training samples, which are put back again to form a number of similar training subsets; it then trains several base classifiers by using these training subsets (MLP classifier). Finally, the classification attributes of samples are determined by the classification results of multiple base classifiers.

4. Experiment setup and performance evaluation

In this section, we summarize our experimental results to detect intrusions using the Self-Organizing map optimized with GA over the KDD'99 datasets. All experiments were conducted on a calculation station 24 CPU Intel Core 2.13GHz, 48GB RAM, running under Linux CentOS 7. The implementation was coded using the Java language.
For a binary classifier, confusion matrix (table 3) shows the results of the real class compared to the predicted result. In these experiments, normal connections are positive events while abnormal represent negative events.

Table 3: Confusion matrix

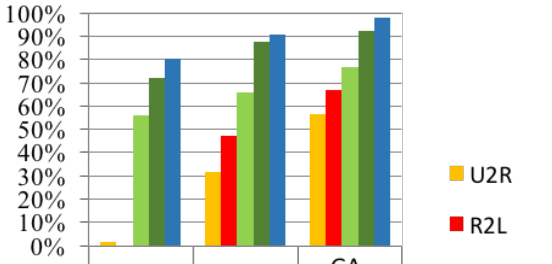
U2R	R2L	Probe	DOS	Normal	
1	1	147	55	9507	Normal
0	39	47	6880	494	DOS
2	38	1862	369	150	Probe
0	1937	63	0	885	R2L
38	0	1	1	27	U2R

To evaluate our approach, we have used four performance indicators from intrusion detection research [19]: True Positive (TP) which is the number of real normal logon events that were correctly classified as normal, False positive (FP) which is the number of abnormal connections of events that have been incorrectly classified as normal connections, False negative (FN) which is the number of normal connections of events that have been incorrectly classified as abnormal connections and finally True negative (TN) which is the number of abnormal connections of events that were correctly classified as abnormal. The performance of the proposed method of intrusion detection was evaluated on all KDDTest+ data set. KDDTrain+ data set was used for training the GA-MLP model. Table 4 illustrates the confusion matrix. This system achieves a top performance of up to 88.77% with a detection rate of 86.21% and only 2.10% of false positive rate.

Table 4: Standard metrics for intrusion system evaluation

Intrusion	**Normal**	
FN=204	TN=9507	**Normal**
TP=10977	FP=1756	**Intrusion**

To validate the prediction results, we compare the proposed technique (GA-Bagging-MLP) with GA-MLP and MLP model trained with the same learning data set. Figure 2 shows the detection rate classified by attack and compares the results of the three models. Note that the results of the proposed model GA-Bagging-MLP provide better detection accuracy. The proposed algorithm has detected 92.32% of DOS attacks whom are the most used by hackers. For Probe attacks, a rate of 76.91% is correctly classified, 67.14% for R2L attacks and 56.72% for U2R.



	MLP	GA-MLP	GA-Bagging-MLP
U2R	1,49%	31,54%	56,72%
R2L	0,00%	47,18%	67,14%
Probe	56,26%	65,97%	76,91%
Dos	72,12%	87,65%	92,23%
Normal	80,33%	90,81%	97,90%

Fig. 2 Multilayer Perceptron neural-network architecture

The obtained result of Ga-Bagging-MLP based on Intrusion Detection System is compared with other existing technologies that are reported by different authors. As the IDS system is developed for NSL-KDD dataset, its performance is also compared. The comparisons of the accuracy between the methods of [3], [5], [7], [8] , [11] and the proposed method are reported in Table 5. The new proposed method achieves better accuracy than the others methods.

Table 5: Comparison of proposed model with other methods by accuracy

Method	***accuracy***
Proposed method	88,77%
Denoising Autoencoder (2017) [11]	88,65%
Autoencoder (2017) [11]	88,28%
RBF-SVM (2012) [3]	85,19%
GAR-FOREST_binary(2015) [7]	82,39%
LOO-OAR-SVM (2014) [5]	82,38%
BAGGING(Partial Tree) (2015) [8]	78,37%
GAR-FOREST multiclass(2015) [7]	77,26%

5. Conclusion

This paper tries to construct an effective ensemble model for intrusion detection system based on Multilayer Perceptron neural network optimized with a Genetic algorithm. The number of the input neurons and the hidden layers of the Multilayer Perceptron neural network is generally the most important parameters to assess because it can produce the best architecture and good performance of prediction. In this work, the Genetic Algorithm is applied to find optimal parameters of Multilayer Perceptron and the accuracy was used as fitness function in the optimization process. Finally, a Bagging ensemble of MLP classifier is formed separately using the best parameters obtained by the optimization process. In the experiments, the NSL-KDD data set is used to learn and evaluate our model. The proposed approach has a major effect on overall accuracy of the analysis (88.77%). The performance of proposed IDS is better than that of other existing machine learning approaches tested on the same dataset.

Acknowledgments

This work is supported in part by the Superior School of Technology, Moulay Ismail University Meknes, which has provided the calculation station where we have executed our experiences.

References

[1] G. Folino, C. Pizzuti, and G. Spezzano, “An ensemble-based evolutionary framework for coping with distributed intrusion detection,” Genetic Programming and Evolvable Machines, vol. 11, no. 2, pp. 131–146, 2010.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “An effective unsupervised network anomaly detection method,” in Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI ’12, 2012, pp. 533–539.

[3] M. Govindarajan and R. Chandrasekaran, “Intrusion detection using neural based hybrid classification methods,” Computer Networks, vol. 55, no. 8, pp. 1662–1671, 2011.

[4] F. Kuang, W. Xu, and S. Zhang, “A novel hybrid KPCA and SVM with GA model for intrusion detection,” Applied Soft Computing Journal, vol. 18, pp. 178–184, 2014.

[5] M. S. Pervez and D. M. Farid, “Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs,” in SKIMA 2014 - 8th International Conference on Software, Knowledge, Information Management and Applications, 2014, pp. 1–6.

[6] G. V. Nadiammai and M. Hemalatha, “Effective approach toward Intrusion Detection System using data mining techniques,” Egyptian Informatics Journal, vol. 15, no. 1, pp. 37–50, Mar. 2014.

[7] N. K. Kanakarajan and K. Muniasamy, “Improving the accuracy of intrusion detection using gar-forest with feature selection,” Advances in Intelligent Systems and Computing, vol. 404. pp. 539–547, 2016.

[8] D. P. Gaikwad and R. C. Thool, “Intrusion Detection System Using Bagging with Partial Decision TreeBase Classifier,” Procedia Computer Science, vol. 49, pp. 92–98, 2015.

[9] A. Saied, R. E. Overill, and T. Radzik, “Detection of known and unknown DDoS attacks using Artificial Neural Networks,” Neurocomputing, vol. 172, pp. 385–393, Jan. 2016.

[10] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, “Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion

detection system," Expert Systems with Applications, vol. 67, pp. 296–303, Jan. 2017.
[11] R. C. Aygun and A. G. Yavuz, "Network Anomaly Detection with Stochastically Improved Autoencoder Based Models," in Proceedings - 4th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2017 and 3rd IEEE International Conference of Scalable and Smart Cloud, SSC 2017, 2017, pp. 193–198.
[12] O. F. Rashid, Z. A. Othman, and S. Zainudin, "A novel DNA sequence approach for network intrusion detection system based on cryptography encoding method," International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 1, 2017.
[13] P. Koprinkova-Hristova, Artificial Neural Networks Methods and Applications in Bio-/Neuroinformatics. 2014.
[14] [14] M. Krawczak, "Multilayer neural networks a generalized net perspective," Studies in Computational Intelligence, vol. 478, pp. 1–194, 2013.
[15] O. Kramer, Genetic Algorithm Essentials, vol. 679. Cham: Springer International Publishing, 2017.
[16] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.
[17] Data set NSL-KDD99." [Online]. Available: http://www.unb.ca/cic/research/datasets/nsl.html. [Accessed: 06-Apr-2018].
[18] S. Aksoy and R. M. Haralick, "Feature normalization `and likelihood-based similarity measures for image retrieval," Pattern Recognition Letters, vol. 22, no. 5, pp. 563–582, 2001.
[19] M. M. M. Hassan, "Current Studies On Intrusion Detection System, Genetic Algorithm And Fuzzy Logic," International Journal of Distributed and Parallel Systems, vol. 4, pp. 35–47, Apr. 2013.

Mehdi Moukhafi received his Master degree in software quality from the faculty of Sciences Dhar El Mahraz, Sidi Mohammed Ben Abdellah University, Morocco and he is currently a PhD Student at the Faculty of Sciences, Moulay Ismail University, Morocco. His fields of interest are machine learning, optimization, security in computer networks.

Khalid El Yassini holds a PhD in mathematics obtained in 2000 with a specialty in Operations Research from University of Sherbrooke in Canada where he obtained a MSc in mathematics in 1994. Previously, he had a degree in applied mathematics (specialty Statistics) at Abdelmalek Essaadi University in Northern Morocco in 1991. During his academic career, he has taught in several Canadian and Moroccan institutions such as University of Sherbrooke, Univesité du Québec à Rimouski, St-Boniface University at Winnipeg-Canada, Moulay Ismail University, International University of Rabat or Royal Military Academy. He has taught various courses in mathematics, computer science, operations research and logistics. Its main current activities focus on mathematical programming including linear programming and multiobjective optimization, artificial intelligence, information systems, security in computer networks, telecommunications, intelligent systems (Smart Cities & Smart Vehicles), logistics engineering, and green logistics with application in hospital field or in medication domain.

Seddik Bri is a Professor at the Electrical Engineering Department in High School of Technology (ESTM), Moulay Ismail University, Meknes -Morocco. His scientific research interests are the microwaves applications and the security in communications systems.