Production Systems and Information Engineering
Volume 8 (2019), pp. 51–67
https://doi.org/10.32968/psaie.2019.004

APPLYING INTRUSION DETECTION ALGORITHMS ON THE KDD-99 DATASET

Mohammad Almseidin
University of Miskolc, Hungary
Department of Information Technology
alsaudi@iit.uni-miskolc.hu

Maen Alzubi
University of Miskolc, Hungary
Department of Information Technology
alzubi@iit.uni-miskolc.hu

Mouhammd Alkasassbeh
Mutah University, Jordan
Information Technology Department
mouhammd.alkasassbeh@mutah.edu.jo

Szilveszter Kovacs
University of Miskolc, Hungary
Department of Information Technology
szkovacs@iit.uni-miskolc.hu

Abstract. Practical task of information reliability and security is the effective intrusion detection and prevention. Open systems are vulnerable. Having in detail information about system structures, more and more sophisticated network intrusion methods could be easily developed and quickly tested. Intruders are always keeping update information about the current technology and generate new intrusion methods. There are several defense solutions against intrusions. The most common solution is Intrusion Detection System (IDS). For giving a short overview of some IDS methods, this paper applies the commonly available KDD-99 dataset for compare and discuss the IDS performance in case of different intrusion types. In this paper, the IDS performance of the J48, Random Forest, Random Tree, Decision Table, Multi-layer Perceptron (MLP) and Naive Bayes Classifier compared based on the average accuracy rate, precision, false positive and false negative performance in case of DOS, R2L, U2R, and PROBE attacks. Moreover, the focus would be on false alarm values. During the tests, the random forest algorithm produced the highest average of accuracy rate 93.77%, while the Random tree algorithm had the lowest rate 90.57%. The lowest value of false negative was produced by the decision table algorithm.

Keywords: KDD-99 dataset, Intrusion Detection, The Denial of service attack, Data mining Algorithms

1. Introduction

Information technology had a rapid development in the last two decades. Computer networks are widely used by industry, business and in various fields of the human life. Maintaining the reliability of networks became an essential task of the IT administrators. On the other hand, the rapid development also produces several challenges and the question of network reliability became a very difficult task. There are many types of attacks threatening the availability, the integrity and the confidentiality of computer networks. The Denial of Service attack (DoS) considered as one of the most common harmful attacks.

The aim of DoS attacks is to temporarily deny services for the end users. In the most common case, it consumes the network resources and overloads the system with undesired requests. For this reason the DoS acts as a large umbrella of naming for all types of attacks which aim to consume computer and network resources. In 2000 Yahoo was the first victim of a DoS attack, which was also the date, when the DoS recorded its first public attack [1]. Nowadays web services and social websites are the main target of DOS attacks [2].

From another vulnerability perspective, the remote to local (R2L) attacks are another common types of attacks which are designed to gain local access permissions remotely in case if some network resources (e.g. servers) are protected by allowing access only for local users. There are several types of R2L attacks e.g. SPY and PHF. These types of attacks aim to prepare illegal remote access to the network resources [3].

Related to the illegal access to the network and computer resources, the type of User to Root (U2R) attacks aim to switch the attacker access permission from normal user to the root user, who has full access rights to the computers and network resources [4]. The main challenge is that attackers are always keeping up-to-date their tools and techniques for exploiting any kind of vulnerabilities appearing to be known. Hence, it is very difficult to detect all types of attacks based on single fixed solutions. For that Intrusion Detection System (IDS) became an essential part of network security. It is designed to monitor the network traffic and generate alerts when any attacks appear. IDS can be implemented to monitor network traffic of a specific device (host IDS) or to monitor all the network traffics (network IDS) which is the most common type used.

Conceptually there are two types of IDS, Anomaly based IDS and Misuse based IDS. Anomaly based IDS implemented to detect attacks based on the recorded normal network behavior. It compares the current real time traffics with the previously recorded normal traffics. This type of IDS is widely used

because it has the ability to detect the new (previously unknown) type of intrusions, too. On the other hand, conceptually it registers the largest values of false positive alarms too, for the situations, which is normal, but not recorded among the "normal network behavior" samples (e.g. there is an uncommonly large number of normal packets considered to be attacking traffic).

Misuse intrusion detection systems are implemented to detect attacks based on a repository of attack signatures. Conceptually it has no false positive alarms but a new type of attack (which signature is missing from the repository) can succeed to pass-through as a normal traffic.

According to [5], attacks detection considered as a classification problem because the target is to clarify whether the packet either a normal or an attack packet. Therefore, an IDS can be built based on the methodology of machine learning algorithms.

To compare the IDS performance of different machine learning algorithms, in this paper, the following algorithms have been studied: J48, Random Forest, Random Tree, Decision Table, Multi-layer Perceptron (MLP) and Naive Bayes Classifier. For the model formation and evaluation the publicly available KDD-99 benchmark dataset was applied. The studied attack types were DOS, R2L, U2R, and PROBE.

The rest of the paper is organized as follows: section (2) summarizes the work related to the IDS application of the KDD-99 dataset and briefly introduces the applied machine learning algorithms. The preprocessing steps of the KDD-99 dataset are discussed in section (3). Section (4) gives a brief overview of the selected data mining algorithms that are used in the experiments. In section (5) some details of the IDS model forming is presented briefly. Section (6) introduces the applied metrics used for evaluating the performance of the IDS methods and discusses the experiments and the achieved results. Finally, section (7) concludes the paper.

2. IDS and the KDD-99 dataset

The commonly available KDD-99 is the data set used at The Third International Knowledge Discovery and Data Mining Tools Competition [6] for the task of building a network intrusion detector. The competition was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining in 1999. Although KDD-99 dataset is rather old, it is still widely used in academic research for testing and comparing IDS performance [7]. Because of its unceasing popularity, for comparing and discussing the IDS performance in case of different intrusion types in this paper also the KDD-99 dataset is chosen. In [2], for a classifier selection model, the authors

made a deep survey of IDS and the KDD-99 dataset. They extracted 49,596 instances of KDD-99 dataset to implement several machine learning algorithms e.g. Naive Bayes and MLP. Authors succeeded to propose two models for detecting intrusions types of the KDD-99 dataset. In [8] the authors applied a MATLAB implementation of Support Vector Machine (SVM) algorithm for IDS. They used the KDD-99 dataset as an IDS benchmark data. They claimed that the SVM algorithm needs long training time and hence the usability of SVM is limited. In [9] the authors preprocessed the KDD-99 dataset, symbolized and normalized the attributes to the [–1, 1] range. Then a feed forward neural network was applied in two experiments. The authors concluded that the neural network is not efficient for detecting R2L and U2R attacks but it has acceptable accuracy rate in detection DOS and PROBE attacks. In [10] the authors are implemented Fuzzy ARTMAP, Radial-basis Function, Back propagation (BP) and Perceptron-back propagation-hybrid (PBH) IDS. The four algorithms evaluated and tested on the KDD-99 dataset, in which the BP and PBH algorithms achieved the highest accuracy rate. Another research direction focuses on attributes selection algorithms in order to reduce the cost of the computation time. In [11] authors are focusing on selecting the most significant attributes to design IDS that have a high accuracy rate with low computation time. They implemented the IDS based on extended classifier and neural network to reduce false positive alarm as much as possible. In [12] the information gain algorithm was implemented to be an effective attributes selection method for improving the DoS intrusion detection. The genetic algorithm (GA) was also implemented to enhance detection of different intrusion types. In [3] the author proposed a methodology to derive the maximum detection rate with the minimum false positive rate. The GA was applied to generate a number of effective rules to detect intrusions. They achieved 97% accuracy on the KDD-99 dataset. In [13] the Naive Bayes algorithm was applied to detect all intrusions types of the KDD-99 dataset. The authors concluded that the detection rate is unacceptable if they apply only a single IDS algorithm. Some IDS research is focusing on a specific type of attack. In [14] a new Distributed Denial of Service (DDoS) dataset is collected from the samples of http ood, smurf, SiDDoS and udp ood attacks data. The DDoS dataset then tested with different IDS algorithms. For detecting the DDoS intrusions, the MLP algorithm achieved the highest accuracy rate (98.36%). Another example for applying the KDD-99 dataset for evaluating different IDS methods can be found in [15], where the performance of 20 different classifiers were compared on different attack categories. Regarding to the implemented experiments the Multivariate Adaptive Regression Splines (MARS) algorithm getting a higher accuracy rate. Furthermore, the fuzzy logic obtained an accepted accuracy rate compared with other implemented algorithms. Moreover,

the lowest accuracy rate recorded by Partial Decision Tree (PART) algorithm. Additionally, The acceptable IDS should perform with an accepted average accuracy rate and lowest possible false negative value.

The KDD-99 dataset still provides a reasonable benchmark environment for testing and evaluating various machine learning algorithms. It is also important to note, that a single machine learning algorithm could not provide an acceptable detection rate. One solution for this problem is the application of different IDS algorithms for detecting various type of attack threats. In the followings seven types of Machine Learning and Data mining Algorithms (J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network) will be implemented, tested, compared and evaluated based on KDD-99 dataset. Our interest is directed to the most important performance parameters, like false negative and false positive attack detections. We would like to select the most promising IDS methods which could achieve an acceptable accuracy rate with the minimum false negative detections.

3. Preprocessing the KDD-99 dataset

The KDD-99 dataset can serve as a good sample for several intrusion behaviors, and good benchmark for testing and evaluating intrusion detection algorithms. The KDD-99 dataset first published by the MIT Lincoln labs at the University of California in 1999 and still available in UCI Machine Learning Archive [16]. It includes 4,898,431 instances with 41 attributes.

The first step of the IDS model generation is the preprocessing of the dataset. For this reason in our case the KDD-99 dataset was first imported to an SQL server 2008, then various statistical measurements values e.g. distribution of instances records, attacks types and occurrence ratios were calculated. Fig. 1 presents the main preprocessing steps of the KDD-99 dataset.

Statistical measurements provide a deep understanding of this dataset in order to extract impartial experiments. Table 1 illustrates the distribution of the attacks types within KDD-99 dataset. There are 21 type of attacks, which can be categorized into four groups with different number of instances and occurrences. 79% of the instances are related to DOS attacks, 19% are belong to normal packets and 2% can be categorized as other attacks types. Based on these values the KDD-99 appears to be an unbalanced dataset. The packets have 41 attributes.

These attributes are basic information which can be collected during the TCP/IP connection [4]. Table 2 illustrates these fundamental TCP/IP attributes. One important contribution of the KDD-99 dataset, that it also contains 32 expert suggested attributes which can help the understanding of

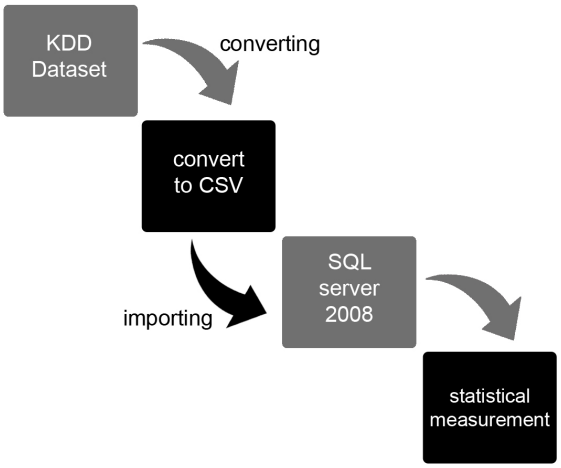

Figure 1. Preprocessing steps of the KDD-99 dataset

the behavior of an attack type. I.e. the most significant attributes of the four attack groups (DOS, R2L, U2R and PROBE) are also included.

4. The applied Data Mining algorithms

This section provides a brief overview of the machine learning algorithms applied for the IDS classification tasks in the rest of the paper. Machine learning algorithms can be categorized as supervised and unsupervised algorithms [17]. Supervised algorithms learn for predicting the object class from pre-labeled (classified) objects. The unsupervised algorithm finds the natural grouping of objects given as unlabeled data. In our IDS study supervised learning algorithms will be applied, as the imported KDD-99 dataset includes predefined classes.

J48 Classifier: This classifier is designed to improve the implementation of the C.4.5 algorithm, which is introduced by Ross Quilan [18] in 1993. The output of this classifier is in the form of decision binary trees, but with more stability between computation time and accuracy then the original C.4.5 [19]. The decision about the expected output is the leaf node of the decision tree structure.

Decision Table Classifier: The main idea of this classifier is to build a lookup table for identifying the predicted output class. There are several algorithms e.g. breadth first search, genetic algorithm and cross validation can be implemented to generate an efficient decision table [20]. The lookup table includes

Table 1. The distribution of the attack types within the KDD-99 Dataset

Categories of Attack	Attack name	Number of instances
DOS	SMURF	2,807,886
	NEPTUNE	1,072,017
	Back	2,203
	POD	264
	Teardrop	979
U2R	Buffer overflow	30
	Load Module	9
	PERL	3
	Rootkit	10
R2L	FTP Write	8
	Guess Passwd	53
	IMAP	12
	MulitHop	7
	PHF	4
	SPY	2
	Warez client	1,020
	Warez Master	20
PROBE	IPSWEEP	12,481
	NMAP	2,316
	PORTSWEEP	10,413
	SATAN	15,892

Table 2. The fundamental attributes of a TCP/IP connection

Attributes	Type
Total duration of connections in second	continuous
Total number of bytes from sender to receiver.	continuous
Total number of bytes from receiver to sender	continuous
Total number of wrong fragments	continuous
Total number of urgent packets	continuous
Protocol type	discrete
Type of service	discrete
The status of the connection (normal or error)	discrete
Label (1) if the connection established from to the same host. Otherwise label (0)	discrete

a set of conditions and the expected classes. These are the rules of the decision table classifier, which are predicting the classes for the incoming inputs [21].

The rules of the decision table can also be fuzzyfied, this case the Decision Table Classifier can also handle uncertainties of the inputs and classes.

Multi-layer Perceptron (MLP) Classifier: MLP is one of the most common algorithms that proved its effectiveness to deal with several application areas e.g. time series classification and regression problems [22]. During the implementation the testing phase can be short, but the training phase typically needs a long time. MLP algorithm can be implemented with various transfer functions e.g. Sigmoid, Linear and Hyperbolic. During the implementation the number of outputs, or expected classes is straightforward, but the number of the hidden layer neurons should be correctly defined for having an effective MLP classifier. At the beginning, every node within the neural network had its randomly weight and bias values, the large weight values in the input layer present the most effective attributes within a dataset, and on the contrary, the small weight values present the least effective attributes within a dataset.

Naive Bayes Classifier: This classifier refers to the group of probabilistic algorithms. It implements Bayes theorem for classication problems. The first step of Naive Bayes classifier is to determine the total number of classes (outputs) and calculate the conditional probability for each dataset classes. After that, the conditional probability is calculated for each attribute. The standard formula of Naive Bayes can be found e.g. in [10]. Furthermore, it has the ability to work with discreet and continuous attributes too. On the contrary of MLP classifier Naive Bayes can be implemented within a short period of time [13]. The Naive Bayes Classifier can be represented as a Bayesian Network (BN) or a Belief Network. BN presents independent conditional probabilities based on understanding framework. In general BN is an acyclic graph between expected class (output) and a number of attributes [23].

Random Tree Classifier: It is one of the classification tree algorithms. The random tree classifier is a finite group of decision trees. The number of trees must be fixed in advance. Each individual tree represents a single decision tree. Each individual tree has randomly selected attributes from the dataset. The entire dataset is predicted from several decision trees outputs and choose the winner expected class based on total numbers of votes [24].

Random Forest Classifier: It is one of the ensemble learning algorithms. The main goal of this algorithm is to enhance trees algorithms based on the concept of the forest. Random forest algorithms [25] have an acceptable accuracy rate. It can be implemented to be able to handle noise in the dataset. It is averaging multiple decision trees, trained on different parts of the same dataset. The number of trees must be fixed in advance. Each individual tree within a forest predicts the expected output. Then the expected output selected by a voting technique [25].

5. Generating the IDS Models

There are 21 types of attacks appearing in the KDD-99 dataset. These attacks are categorized into four groups (DOS, R2L, U2R, and PROBE). Each attack types has different number of instances and occurrences in dataset.

After the preprocessing of the KDD-99 dataset, 148,753 instances of records have been extracted to an SQL server. This labeled data serves as a training set for the further IDS model generation. The attack categories and types with the number of instances are presented on Table 3. Based on the analysis of KDD-99 dataset the occurrence distribution of the different attack types was recorded. 79% of the extracted data present DOS attacks, 19% is related to the instaces of normal traffic and 2% is related to other types of intrusions (U2R, R2U and PROBE).

Table 3. The Training Model Dataset.

Categories of Attack	**Attack name**	**Number of instances**
DOS	SMURF	85,983
	NEPTUNE	32,827
	Back	70
	POD	10
	Teardrop	30
U2R	Buffer overflow	10
	Load Module	2
	PERL	1
	Rootkit	5
R2L	FTP Write	2
	Guess Passwd	10
	IMAP	4
	MulitHop	2
	PHF	1
	SPY	1
	Warez client	31
	Warez Master	7
PROBE	IPSWEEP	382
	NMAP	70
	PORTSWEEP	318
	SATAN	487
Normal		28,500

In this paper, the experiments were performed on an Ubuntu 13.10 platform, Intel R, Core(TM) i5-4210U CPU @ 1.70GHz (4CPUs), 6 GB RAM. The

applied machine learning tool was the Waikato Environment for Knowledge Analysis (WEKA) [26]. It is an open source tool written in JAVA and available for free. It provides all the classifiers referred in this paper. These are the J48, Random forest, Random Tree, Decision Table, Multilayer Perceptron (MLP), Naive Bayes and Bayes Network. Based on the preprocessed 148,753 instances according to the labeled attack categories (see attack types and categories on Table 3) all the seven studied classifiers were created. For the creation of the classifiers all the labeled data were processed as training data. The cross validation of the classifiers were omitted, as our goal was to compare the best available performance of the different type of classifiers based on an existing data (KDD-99 records), not on an unknown data set. Then the classifiers were saved for a comprehensive study introduced in the followings.

6. Performance of the IDS implementation

After the IDS model generation, the next step is the comparative study of the models. In order to implement a fair testing phase fully randomized 60,000 instances have been extracted from the preprocessed database. The extracted testing data included all the 21 attack types of the KDD-99 dataset and labeled according to the attack categories introduced on Table 3. There are several metrics that can be used for evaluating the efficiency of the IDS model. In this paper, the confusion matrices were generated for each classification algorithms. Furthermore, the following performance metrics [14] were computed:

- **True Positive (TP)**: This value represents the correct classification of the attack packets as attacks.
- **True Negative (TN)**: This value represents the correct classification of the normal packets to be normal traffic.
- **False Negative (FN)**: This value represents an incorrect classification, where the attack packet classified as normal packet. A large FN value presents a serious problem of confidentiality and availability of network resources, because the attacker succeeded to pass through the IDS.
- **False Positive (FP)**: This value represents incorrect classification, where the normal packet classified as an attack. The increasing of FP value increases the computation time, but it is considered as less harmful than the increased FN value.
- **Precision**: Is one of the primary performance indicators. It presents the total number of records that are correctly classified as attack divided by a total number of records classified as attack. The precision can be calculated as follows:

$$P = \frac{TP}{(TP + FP)} \tag{6.1}$$

In addition, the number of both correctly and incorrectly classified instances are recorded with respect to the time taken for proposed training model.

During the testing phase, the following classification parameters were applied:

- **J48 tree classifier**: confidence factor = 0.25; numFolds = 3; seed = 1; unpruned = False, collapse tree = true and sub tree rising = true.
- **Random forest classifier**: number of trees = 100 and seed = 1.
- **Random tree classifier**: min variance = 0.001 and seed = 1.
- **Decision Table classifier**: Best First Search (BFS) and cross value = 1.
- **MLP classifier**: search learning rate = 0.3, momentum = 0.2, validation threshold = 20.

Table 4 presents the TP rate and the Precision values of the studied classification algorithms during the experiments. It can be concluded that the random forest classifier achieved the highest 93.1% TP rate, and the random tree classifier achieved the lowest 90.6% TP rate. I.e. the random tree classifier has the lowest correct attacks classification value. The decision table classifier reached the lowest 94.4% precision value. This indicates that the decision table classifier suffers from an increasing false positive value. Therefore, there is a large number of normal packets classified as attack packets.

Table 4. The True Positive Rate and the Precision

Classification Algorithms	**TP Rate**	**Precision**
J48	0.931	0.989
Random forest	0.938	0.991
Random tree	0.906	0.992
Decision table	0.924	0.944
MLP	0.919	0.978
Naive Bayes	0.912	0.988
Bayes Network	0.907	0.992

In general, the TP rate and precision values are important performance parameters for a common intrusion detection system, but from another perspective the most serious performance parameters are the FP rate and the FN rate. The goal of this study is to decrease both of these parameters, as much as possible, especially the FN parameters. The FP and FN performance

parameters of the IDS tests are summarized on Table 5. It can be concluded, that the random tree classifier achieved the highest 0.093 FN rate. Hence there is a large number of attacks classified as normal packet. On the contrary with the decision table classifier which is achieved the lowest 0.002 FN rate. In the same time, the decision table classifier reached the highest 0.073 FP rate. It means that there is a large number of normal packet classified as attack packets.

Table 5. The False Positive Rate and the False Negative Rate

Classification Algorithms	**FP Rate**	**FN Rate**
J48	0.005	0.063
Random forest	0.001	0.061
Random tree	0.001	0.093
Decision table	0.073	0.002
MLP	0.014	0.066
Naive Bayes	0.002	0.085
Bayes Network	0.001	0.092

Table 6 presents the Root Mean Square Error (RMSE) and area under the Receiver Operating Characteristic (ROC). RMSE presents the difference between the actual and the desired outputs based on the confusion matrix. The model with lower value of RMSE indicates better output prediction efficiency, on the contrary large value of RMSE indicates lower prediction efficiency. The ROC value is calculated based on the true positive and the false positive values. The large value of ROC indicates that the model has better intrusion detection ability, while the lower value present the weakness of the model.

Table 6. The Root Mean Square Error and the Area under the Receiver Operating Characteristic

Classification Algorithms	**ROC Area**	**Root Mean Squared Error**
J48	0.969	0.0763
Random forest	0.996	0.0682
Random tree	0.953	0.0763
Decision table	0.984	0.0903
MLP	0.990	0.0813
Naive Bayes	0.969	0.0872
Bayes Network	0.997	0.0870

According to the results on Table 7, the Bayes network classifier achieved the highest 0.997 ROC value, while the random tree classifier achieved the lowest 0.953 value. Furthermore, the random forest classifier had the lowest 0.0682 RMSE value, while the decision table presented the highest 0.0903 value. After the classification of 60,000 instances of the KDD-99 dataset, the total number of incorrectly classified records for each selected classifier, and the average accuracy rate is presented on the Table 7. The average accuracy rate is calculated according to the following formula:

$$AverageAccuracyRate = \frac{TP + TN}{TP + FN + FP + TN} \tag{6.2}$$

Table 7. Average Accuracy Rate

Classification Algorithms	**Correctly classified Instances**	**incorrectly classified Instances**	**Accuracy Rate**
J48	55,865	4,135	93.10%
Random Forest	56,265	3,735	93.77%
Random tree	54,345	5,655	90.57%
Decision table	55,464	4,536	92.44%
MLP	55,141	4,859	91.90%
Naive Bayes	54,741	5,259	91.23%
Bayes Network	54,439	5,561	90.73%

It is important to mention, that it could take a long time to build the IDS model. Based on the experiments, the building the random tree classifier model is the fastest, while training the MLP classifier was taken about 176 minutes. In our experiments it was the longest model generation time. From the results of the tests, we can conclude the followings:

- **The Random forest** achieved the highest 93.77 accuracy rate with the smallest RMSE value and false positive rate.
- **The Random tree** classifier reached the lowest 90.73 average accuracy rate with smallest ROC value.
- **Regarding to** the average accuracy rate there is no big difference between the MLP classifier and the Naive Bayes classifier.
- **All classification** algorithms present acceptable precision rates for detecting normal packets.
- **Bayes network** classifier recorded the highest value for detecting correctly the normal packets.

- **There are no** big differences between the MLP and the J48 algorithms based on FN parameters.
- **Despite** that the decision table classifier did not reached the highest accuracy rate, but it had the lowest FN rate. The model generation time was also acceptable.
- **All of the tested** classification algorithms had acceptable model generation time, except the MLP.
- **It can be concluded** that the rule based algorithms (decision table) are presented an acceptable accuracy rate with the lowest FN rate, which is increasing the confidentiality and the availability of the network resources.

7. Conclusions and Future Works

The KDD-99 dataset was applied for measuring the performance of seven classification algorithms (Random tree, Random forest, Naive Bayes, MLP, Decision table and J48) in IDS performance. The KDD-99 dataset includes instances from 21 types of attacks from four attack groups (DOS, R2L, U2R, and PROBE). Each attack types has different number of instances and occurrences in dataset. In our tests first the IDS models for the seven studied classification algorithms were generated. Then their IDS performances were tested based on randomly chosen KDD-99 data.

According to our experiments, from 60,000 randomly chosen testing records, the random forest algorithm achieved the highest 93.77% accuracy value. During the same test it has 3,735 incorrectly classified records. The random tree algorithm achieved the lowest 90.57% accuracy value with 5,655 incorrectly classified records. Regarding to the root mean squared error values, also the random forest algorithm achieved the lowest 0.0682 value, while the decision table algorithm had the highest 0.0903 value. The Naive Bayes algorithm needed the shortest model generation time, while the MLP algorithm reached the longest 176 minute training time.

All the seven studied classification algorithms achieved acceptable precision for detecting normal packets. The decision table algorithm had the lowest 0.002 false negative value, which means that it can detect various intrusion types of the KDD-99 dataset successfully. The effectiveness of any IDS always suffers from false negative values. The acceptable IDS should perform with the lowest possible false negative value. Consequently, as a part of the future work, we would like to modify the rule based decision table algorithm to a fuzzy rule based system to generate an IDS model which can achieve an acceptable accuracy rate with the lowest possible false negative classification.

8. Acknowledgement

The described study was carried out as part of the EFOP-3.6.1-16-00011 "Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialization" project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] G. C. Kessler, "Defenses against distributed denial of service attacks," *SANS Institute*, vol. 2002, 2000.

[2] H. A. Nguyen and D. Choi, "Application of data mining to network intrusion detection classifier selection model," in *Challenges for Next Generation Network Operations and Service Management: 11th Asia-Pacific Network Operations and Management Symposium, APNOMS 2008, Beijing, China, October 22-24, 2008. Proceedings*, vol. 5297. Springer Science & Business Media, 2008, p. 399.

[3] S. Paliwal and R. Gupta, "Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57–62, 2012.

[4] M. Tavallaee, E. Bagheri, W. Lu, and 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, pp. 1–6. [Online]. Available: https://doi.org/10.1109/cisda.2009.5356528

[5] P. Amudha, S. Karthik, and S. Sivakumari, "Classification techniques for intrusion detection-an overview," *International Journal of Computer Applications*, vol. 76, no. 16, 2013. [Online]. Available: https://doi.org/10.5120/13334-0928

[6] "Kdd cup 1999 data," 1999, accessed on 01.07.2017. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[7] A. Ozgur and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, no. e1954v1, 2016. [Online]. Available: https://doi.org/10.7287/peerj.preprints.1954v1

[8] M. K. Lahre, M. T. Dhar, D. Suresh, K. Kashyap, and P. Agrawal, "Analyze different approaches for ids using kdd 99 data set," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 1, no. 8, pp. 645–651, 2013.

[9] F. Haddadi, S. Khanchi, M. Shetabi, and V. Derhami, "Intrusion detection and attack classification using feed-forward neural network," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. IEEE, 2010, pp. 262–266. [Online]. Available: https://doi.org/10.1109/iccnt.2010.28

[10] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in *Proc. IEEE Workshop on Information Assurance and Security*, 2001, pp. 85–90.

[11] W. Alsharafat, "Applying artificial neural network and extended classifier system for network intrusion detection." *International Arab Journal of Information Technology (IAJIT)*, vol. 10, no. 3, 2013.

[12] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision tree analysis on j48 algorithm for data mining," *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013.

[13] C. Fleizach and S. Fukushima, "A naive bayes classifier on 1998 kdd cup," 1998.

[14] M. Alkasassbeh, G. Al-Naymat, A. B. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *International Journal of Advanced Computer Science & Applications*, vol. 1, no. 7, pp. 436–445. [Online]. Available: https://doi.org/10.14569/ijacsa.2016.070159

[15] S. O. Al-mamory and F. S. Jassim, "Evaluation of different data mining algorithms with kdd cup 99 data set," *Journal of Babylon University/Pure and Applied Sciences*, vol. 21, no. 8, pp. 2663–2681, 2013.

[16] S. D. Bay, "The uci kdd archive [http://kdd. ics. uci. edu]. irvine, ca: University of california," *Department of Information and Computer Science*, vol. 404, p. 405, 1999.

[17] M. Al-Kasassbeh, "Network intrusion detection with wiener filter-based agent," *World Appl. Sci. J*, vol. 13, no. 11, pp. 2372–2384, 2011.

[18] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[19] M. S. Bhullar and A. Kaur, "Use of data mining in education sector," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2012, pp. 24–26.

[20] R. Kohavi and D. Sommerfield, "Targeting business users with decision table classifiers." in *KDD*, 1998, pp. 249–253.

[21] P. Aditi and G. Hitesh, "A new approach of intrusion detection system using clustering, classification and decision table," 2013.

[22] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on neural networks*, vol. 3, no. 5, pp. 683–697, 1992. [Online]. Available: https://doi.org/10.1109/72.159058

[23] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[24] A. Cutler and G. Zhao, "Pert-perfect random tree ensembles," *Computing Science and Statistics*, vol. 33, pp. 490–497, 2001.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009. [Online]. Available: https://doi.org/10.1145/1656274.1656278