

## Review of Computer Engineering Research

2019 Vol. 6, No. 1, pp. 12-23

ISSN(e): 2410-9142


ISSN(p): 2412-4281


DOI: 10.18488/journal.76.2019.61.12.23

© 2019 Conscientia Beam. All Rights Reserved



## CLASSIFICATION ENSEMBLE BASED ANOMALY DETECTION IN NETWORK TRAFFIC

 **Ramiz M. Aliguliyev**<sup>1</sup>

 **Makrufa Sh. Hajirahimova**<sup>2+</sup>

<sup>1,2</sup>*Institute of Information Technology of Azerbaijan National Academy of Sciences, B.Vahabzade Str. 9A, Baku, AZ1141, Azerbaijan*

<sup>1</sup>*Email: [r.aliguliyev@gmail.com](mailto:r.aliguliyev@gmail.com)*

<sup>2</sup>*Email: [makrufa@science.az](mailto:makrufa@science.az)*



(+ Corresponding author)

### ABSTRACT

#### Article History

Received: 5 November 2018

Revised: 24 December 2018

Accepted: 29 January 2019

Published: 14 March 2019

#### Keywords

Anomaly detection

Big data analytics

Network security

An ensemble of classifiers

IDS

Denial of service.

Recently, the expansion of information technologies and the exponential increase of the digital data have deepened more the security and confidentiality issues in computer networks. In the Big Data era information security has become the main direction of scientific research and Big Data analytics is considered being the main tool in the solution of information security issue. Anomaly detection is one of the main issues in data analysis and used widely for detecting network threats. The potential sources of outliers can be noise and errors, events, and malicious attacks on the network. In this work, a short review of network anomaly detection methods is given, is looked at related works. In the article, a more exact and simple multi-classifier model is proposed for anomaly detection in network traffic based on Big Data. Experiments have been performed on the NSL-KDD data set by using the Weka. The offered model has shown decent results in terms of anomaly detection accuracy.

**Contribution/Originality:** This study proposed multi-classifier model for increasing anomaly detection accuracy in network traffic. The model consists of the J48, LogitBoost, IBk, AdaBoost, RandomTree classifiers. This work performed a comparative analysis of used classifiers and their combination to see which one will give the best result. In study classifiers and their combination have been implemented on NSL-KDD open source dataset using WEKA tool. The results show that the ensemble classifiers provide the better result than using these classifiers individually. The computer network traffic analysis with employment of our model can help network engineers and administrators to create a more reliable network, avoid possible discharges and take precautionary measures.

## 1. INTRODUCTION

Security issues have always been thought of people from the existence of humanity. In the Big Data era, the interest in security issues has also increased and has turned into a very serious scientific-research direction in some aspects as political, economic, social, demographic, military, environmental, and so on. As shown in Hajirahimova [1] can be seen from two different aspects Big data and information security issues: Application of Big Data Analytics in Information Security and Information Security Problems in Big Data Technologies. In other words, these approaches are two sides of the same coin. Therefore we must use intelligent analytics parallel to traditional security mechanisms in improving Big Data security [2].

Because of increased network connectivity, computer systems are becoming increasingly sensitive to attack. Widely used by network services, web additions has pointed out the implementation of security measures against network and computer threats in enterprises and organizations. So, lately a number of cyber-targeted attacks as cyber-terrorism, cyber-wars, APT- Advanced Persistent Threat is rapidly increasing [3]. Collecting large volume of data from network, host, security devices, etc. to detect these attacks, which greatly threaten the security of corporate computers updating the issue of detecting anomalies in the data and requires more effective analysis methods and algorithms for solving the problem. Because early detection of dangerous traffics on computer networks, analysis of log files is an essential condition in providing network security. The detection of anomalies is one of the main issues in data analysis. Anomalies detection allows you to interfere with unusual behaviors. Precautionary prevention of attacks (from 0 the day), filtering of previously unknown and malicious data is particularly important.

An anomaly is understood as a regularity that is not consistent with normal behavior in the data or defined the indicator of the data. In studies, "outliers", "exceptions", "peculiarities", "surprise" terms are also used as a synonym for the anomaly [4]. In other words, the detection of anomalies is the problem of finding templates that are not appropriate for probability acceptable behavior. This problem is more noticeable in the context of big data. Traditional methods for detecting anomalies do not show good results in the big data which determined features such as large volumes, variations, high speeds [5-8].

Inaccurate detection or processing of anomalies direct impacts on the reliability of gained knowledge. Therefore, proper identification of anomalies is an important issue, but also not a simple matter.

In this context, Google's MapReduce framework provides an effective method for analysis of large amounts of data, Technologies such as Clouds Computing, MapReduce, Hadoop, etc., have enough computing power to handle large-scale data processing. With the helping of these technologies, it has been possible to integrate and analyze multiple network data. As a result, security analytics technology has been acquired based on big data analysis [1, 6, 9].

The purpose of the submitted study is analyzing anomalies more accurately, more precisely based on Big Data Analytics. Big data analytics is the process of detecting hidden didactics, unknown correlation and other useful information in large volumes of data for making optimal (the best) decision. In this context, anomaly detection is in the focus of attention both in scientific research and application fields as a very serious problem [1, 2, 9-11]. Detection of anomalies is one of the application contexts in the large-scale data at various information security objects. For this purpose, in this paper, a model is offered for detecting network attacks based on Big Data Analytics. In the proposed model, the object of the research is network traffic that is considered one of the Big Data sources (log files that stored on the server for analysis and monitoring data, etc.).

The next parts of the paper have been organized as follows: In the second section of the article existing approaches are discussed in network traffic anomaly detection. In the third section, the proposed model has been described. In the fourth section experiments and results are interpreted. Later conclusion and a list of used literature are given.

## 2. RELATED WORK

Generally, Intrusion Detection Systems (IDS) are the first systems based on intelligent data analysis in the field of information security [12-14]. The purpose of IDS is to detect malicious traffic – anomalies by controlling both input and output traffic. That's IDS can classify events and behaviors as malicious and normal. These systems are usually based on rules (or signature-based IDS) and anomalies. The detection systems of traditional threats based on rules can find known templates. The main problem is that the majority of errors (classification or reversal of harmful flows in traffic) and inability to use high-capability networks, failure to identify new attacks. Anomalies-based approaches are able to detect distinctive templates than normal behaviors. This is its advantage.

The detection of anomalies in the data study has begun in the nineteenth century [4]. Over time, special methods have been developed to detect anomalies in many application fields. Some methods consist of more general theoretical approaches. In Chandola, et al. [4]; Hodge and Austin [13]; Wang, et al. [15] is given a wide overview of anomalies detection methods. In these studies are given various aspects of anomaly detection problems (nature of data, types of anomalies - context anomaly, collective anomaly, data label), classification of anomaly detection based on methods.

From a methodological point of view network anomaly detection methods are divided into two groups: stochastic and deterministic. In stochastic methods data is modeled according to probability. Stochastic methods adapt data to a predictable model and assess the compatibility of new traffic compared to this model. Evaluation is based on a statistical hypothesis. Deterministic methods divide the function into two parts: "normal" and "abnormal". Borders are defined by cluster analysis and SVM methods, etc. In terms of data, the anomaly detection methods are flow-based, packet-based and window-based [16].

Studies show that network detection methods are divided into two types: supervised and unsupervised. Under supervised detection methods, the normal behavior model of the system or network is created based on the training data. In unsupervised detection methods is not used any training data [4, 13, 16].

In numerous studies, statistical approaches for detection of anomalies reduction of size, based on machine learning, neural networks, the Bayesian network, entropy, based on rules and optimization, SVM-based, etc. models and algorithms were proposed [8, 10, 16, 17].

The analysis of network traffic can help engineers to create a reliable network, be protected from extra downloads, predict dangers in advance. For this purpose, packet size, duration, IP addresses, ports and etc. usually has been used in research.

In Adibi [18] Adibi has commented on the essence of traffic classification methods in the context of Packet, Flow, and Application.

In many studies, hybrid or multi-level classification models have been suggested to increase the accuracy of classification in the detection of anomaly [8, 10, 19-22]. The combined use of numerous data mining methods is known as an ensemble approach, and the process of learning the correlation between these ensemble techniques is known by names such as meta-learning.

The researchers of Columbia University propose using of mechanism that consists of some classifiers for increasing accuracy and efficiency of IDS. They show that a method cannot detect an attack, the probability of detecting an attack of other method is high [23].

A Branitskiy and Kotenko have proposed hybridization model of intelligent computation methods as neural networks, neuro-fuzzy classifiers, and SVM with the purpose of effective detection of network attacks. Authors use the method of principal components to accelerate the processing of input vectors. A multi-level analysis of network traffic is one of the advantages of the proposed model [22].

In Aljawarneh, et al. [21] Aljawarneh and others propose a hybrid algorithm as consisting of J48, MetaPaging, Random Tree, REPTree, AdaBoostML, DecisionStump, and NaïveBayes classifiers, that is measured with high accuracy degree and allows minimization of both computation and time.

In Imamverdiyev and Sukhostat [24] approach was proposed which is based on informative features for anomaly detection of network traffic. A higher dimension characterized by the number of features which is the main problem in the analysis of network traffic. The selection (reduction) of main features improves the efficiency of classification. Reduction of feature also allows interpreting results better.

In cybersecurity field majority of current methods consists of heuristic approaches which have high computation complexity. In Pajouh, et al. [19] machine learning approach is proposed to detect anomalous traffic in the network. Proposed two-tier classification model provides high detection speed as a result of the optimal reduction of dimension. It allows detecting less common like U2R, R2L, and more dangerous attacks exactly.

In Reháková, et al. [25] a specific model is proposed that based on trust and reputation mechanism to detect network anomaly by the integration method of some algorithms. In the network, the model can identify the important events (scans, DoS attacks, worms etc.) reliably. Authors get reducing errors, false positives, and false negatives by applying of agent technology to the analysis of network behavior. So, they get more exact results.

The researchers of Purdue University propose a new architecture model to define anomalous behaviors of the system by using Genetic Programming for detection of attacks [26].

Networks are complex interacting systems and are comprised of several items such as routers and switches. Researchers have approached the anomaly detection problem in networks using various techniques such as artificial intelligence, machine learning etc. Greater synergy between the networking and signal processing areas will help develop better and more effective tools for detecting network anomalies problems. The application of signal processing techniques to this area is still in its infancy, and we believe that it has great potential to enhance the field, and thereby improve the reliability of IP networks. In this paper, authors review the use of signal processing techniques to address the problem of measuring, analyzing, and synthesizing network information to obtain normal network behavior [27].

### 3. PROPOSED MODEL

#### 3.1. Statement of the Issue

In the article multi-classification approach that allows detecting anomaly in the network traffic network is suggested in Figure 1. Let's assume that the  $D = \{x_1, x_2, \dots, x_n\}$  dots majority has been given. Classification algorithms in quantity  $M = \{m_1, m_2, \dots, m_k\}$  have been selected. High-precision detection of harmful traffic in the network is required to increase the efficiency of IDS.

The proposed algorithm presents the evaluation vector of the classifier for every data point in the dataset.

$$A = (a_{ij})_{n \times k} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{pmatrix}$$

Where,  $n$  – is the number of data points in the dataset,

$k$  – is a number of classifiers, that takes part in the ensemble,

$a_{ij}$  – is an evaluation of classifier for every data point in the dataset.

The development of the proposed model consists of the following steps:

*Step 1.* Selection of training and test datasets-NSLKDD. About this dataset will be informed in the next sections;

*Step 2.* First processing phase. Noisy data clearance for the purpose of only storing useful information or the application of normalization or correction methods to simplify the processing process;

*Step 3.* Build a hybrid model consisting of 3.J48, LogitBoost, IBk, AdaBoost, RandomTree classifiers;

*Step 4.* Testing of classifiers on data;

*Step 5.* Choosing a method for creating a classifier ensemble (for example, Stacking).

As shown in the scheme the proposed method consists of two phases for anomaly detection. At the first phase training is performed, and at the second phase algorithm is tested for anomaly detection.

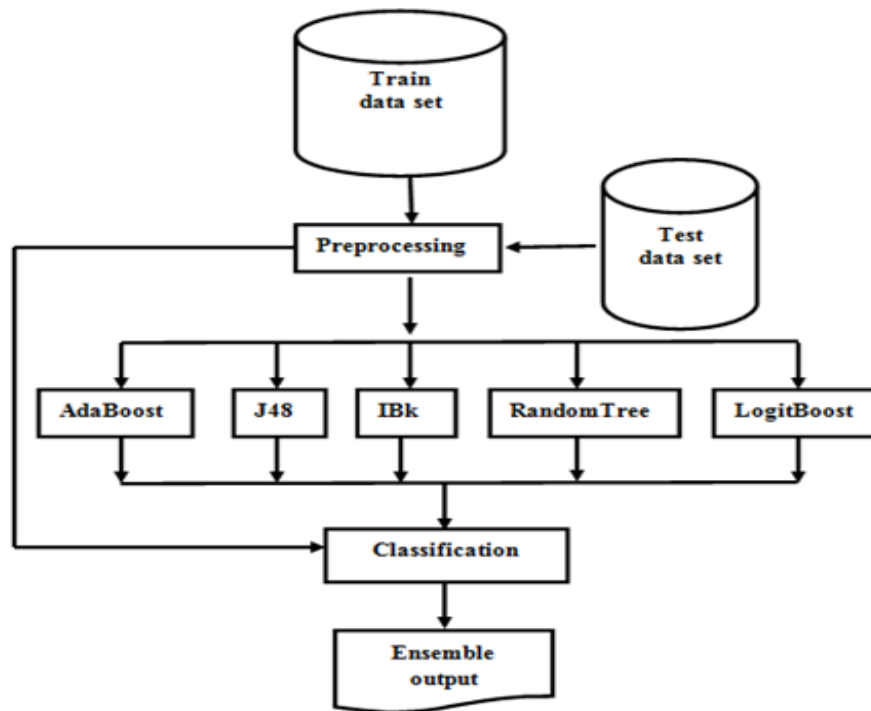


Figure-1. Scheme of the proposed model.

#### A. Dataset – NSL KDD

One of the main problems faced by researchers in detecting network anomalies is the lack of open data sets. In order to solve the problem, the importance of etalon test data for the testing of IDS systems in the 1990s was recognized and the DARPA data collectors appeared [19, 28].

It should be noted that at present, many data sets are available to the researchers for testing IDS systems: DARPA, KDD'99, Internet Traffic Archive, LBNL, CAIDA, DEFCON, PREDICT, ISCX 2012 and so on Mohiuddin, et al. [28].

DARPA data sets have been created with the imitation of observed network traffic in mid-range USA airbase for network security purposes. The imitated attacks on the KDD training dataset are divided into four categories [19, 20, 28]:

- Denial attack from Service or Denial of Service (DoS) - the attacker is overloading certain services that are abandoned service to legitimate users.
- User to Root Attack (U2R) - the attacker tries to gain access to the administrative account from normal user account using certain exploits.
- Remote to Local Attack (R2L) - the attacker using certain holes in the computer attempts to access the local user account on that computer.
- Probing Attack: The Probe attack attempts to collect information about a computer network to break information security.

DARPA and KDD data collections are considered outdated, have a number of shortcomings, but are still being used. An improved version of KDD'99 - NSL-KDD data collection has been developed to eliminate deficiencies [29]. In the training collection, there are no unnecessary records, no repetitions in the test collection, and so on. Taking this into account, in the paper are used NSL-KDDTrain dataset that consists of 125973 records and NSL-KDDTest data consisting of 22544 records.

Table-1. List of attributes for each NSL-KDD database record.

No	Attribute name	No	Attribute name
1.	duration	22.	is_guest_login
2.	protocol_type	23.	count
3.	service	24.	srv_count
4.	flag	25.	error_rate
5.	src_bytes	26.	srv_error_rate
6.	dst_bytes	27.	error_rate
7.	land	28.	srv_error_rate
8.	wrong_fragment	29.	same_srv_rate
9.	urgent	30.	diff_srv_rate
10.	hot	31.	srv_diff_host_rate
11.	num_failed_logins	32.	dst_host_count
12.	logged_in	33.	dst_host_srv_count
13.	num_compromised	34.	dst_host_same_srv_rate
14.	root_shell	35.	dst_host_diff_srv_rate
15.	su_attempted	36.	dst_host_same_src_port_rate
16.	num_root	37.	dst_host_srv_diff_host_rate
17.	num_file_creations	38.	dst_host_error_rate
18.	num_shells	39.	dst_host_srv_error_rate
19.	num_access_files	40.	dst_host_error_rate
20.	num_outbound_cmds	41.	dst_host_srv_error_rate
21.	is_host_login	42.	class

Source: NSL-KDD [30], Dhanabal and Shantharajah [31]; Revathi and Malathi [32]

Each record examples in the NSL-KDDTrain dataset contains 42 attributes Table 1 that are marked as normal or anomalies and reflect different properties in themselves. It should be noted, TCP, UDP, and ICMP protocols have been used in NSL- KDD dataset. In NSL-KDD [30]; Dhanabal and Shantharajah [31]; Revathi and Malathi [32] detailed information about data, attributes, names, descriptions and so on were informed.

### B. WEKA (Waikato Environment for Knowledge Analysis)

It is more important to have the perfect tools for the intelligent analysis of data. The first version of Weka's open source software was developed in Java programming language at Waikato University in New Zealand in 1993, to provide data analysis and machine learning algorithms. This allows it to be used on any computer platform. WEKA offers researchers with initial processing tools, multiple classification and clustering, regression methods and provides visualization of results.

Over the past years, the software has been developed, and researchers have been created the most up-to-date opportunities. It should be noted that the application of latest version 3.8.1 to the Big Data has been realized<sup>1</sup>.

### C. Classification Methods

**Decision Tree (or J48)** is predictive machine learning language that it decides the target value of new example based on different features of available information. Decision trees create a hierarchical partitioning of the data, which relates the different partitions at the leaf level to the different classes. The hierarchical partitioning at each level is created with the use of a split criterion. The split criterion may either use a condition (or predicate) on a single attribute, or it may contain a condition on multiple attributes. The overall approach is to try to recursively split the training data so as to maximize the discrimination among the different classes over different nodes [33]. In Weka implemented an algorithm for creating decision trees under the name J48.

**KNN (or IBk)** – Non-parametric classifier "Nearest neighbor" is also known as "Instance-based" learning. K nearest neighbors are simple classification algorithm based on similarity or distance calculation between instances. To classify an unknown instance represented by some feature vectors as a point in the feature space, the KNN

<sup>1</sup> <https://www.cs.waikato.ac.nz/~ml/weka/>.

classifier calculates the distances between the point and points in the training dataset. For the labeling of each test sample objects must sequentially perform the following operations [34]:

- Calculate the distance to each of the training sample objects;
- Select  $k$  training sample objects, the distance to which is minimal;
- The class of the object being classified is the class most often encountered among the  $k$  nearest neighbors.

Usually, the Euclidean distance function is the most widely used the distance metric.

$$D(x,y)=\sqrt{\sum_{i=1}^k(x_i-y_i)^2}.$$

Some other distance functions are also available for KNN classification, such as Minkowsky, Manhattan etc.

$$D(x,y)=\left(\sum_{i=1}^k(|x_i-y_i|^q)\right)^{\frac{1}{q}}$$

$$D(x,y)=\sum_{i=1}^k|x_i-y_i|$$

The advantages of KNN are its ease of interpretation. However, on real problems, it often turns out to be ineffective. In addition to the classification accuracy, the classification problem is the classification speed: if in the training sample  $N$  objects, in the test selection of  $M$  objects and the dimension of the space is  $K$ , the number of operations for classification of the test sample can be estimated as  $O(K * M * N)$ . And yet, the KNN algorithm is a good example of getting to know Machine Learning.

**The AdaBoost** short for adaptive boosting, algorithm, introduced in 1995 by Freund and Schapire [35] Boosting originated from the question of whether a set of weak classifiers could be converted to a strong classifier. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. That is, it focuses on classification problems and is aimed at converting a set of weak classifiers into a strong one. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems, it can be less susceptible to the overfitting problem than other learning algorithms [36]<sup>2</sup>.

**LogitBoost** is also referring to a boosting classification algorithm. The LogitBoost algorithm is formulated by professors at Stanford University Jerome Friedman, Trevor Hastie, and Robert Tibshirani. LogitBoost and AdaBoost are close to each other in the sense that both perform an additive logistic regression. The difference is that AdaBoost the exponential loss, whereas LogitBoost minimizes the logistic loss <sup>3</sup>.

**A random tree** is a tree or arborescence that is formed by a stochastic process. Types of random trees include Random binary tree, Random recursive tree, Random forest, etc <sup>4</sup>.

#### 4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section carried out experiments carried out and their results are summarized. Experiments were made on a computer with a Windows 8.1 (64bit) operating system, Intel (R) Core (TM) i5-2400 processor, 4GB of RAM, and the training and testing NSL-KDD dataset were used to detect anomalies in network traffic. These files contain

<sup>2</sup> <https://codesachin.wordpress.com/tag/adaboost>.

<sup>3</sup> <https://software.intel.com/en-us/daal-programming-guide-logitboost-classifier>.

<sup>4</sup> [https://en.wikipedia.org/wiki/Random\\_tree](https://en.wikipedia.org/wiki/Random_tree).

42 attributes as well as "protocol\_type", "service", "flag", "src\_bytes", "dst\_bytes", "land", "wrong\_fragment", "urgent", "hot", dst\_host\_count, etc.

In this work J48, LogitBoost, IBk, AdaBoost, RandomTree classifiers were tested according to all feature vectors. As mentioned previously the evaluation of the efficiency of the proposed model was taken on NSL-KDD open source database in the WEKA environment. The Stacking method applies to create classifiers ensemble. SVM Radial Basis Function was used as meta-classifier in classifier ensemble. Ensemble classifier is made up of multiple classifier algorithms and whose output is a combined result of the output of those classifier algorithms.

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. The individual classification models are trained based on the complete training set; then, the meta-classifier is fitted based on the outputs – meta-features – of the individual classification models in the ensemble. The meta-classifier can either be trained on the predicted class labels or probabilities from the ensemble [37]<sup>5</sup>.

#### 4.1. Interpretation of Classification Results

In the result of the classification process confusion matrix is gained. The dimension of the matrix may be two and more according to the number of classes. In the research two-dimensional matrix is obtained for having two classes (normal, anomaly) and its structure is shown in Table 2.

**Table-2.** List of attributes for each NSL-KDD database record

Normal	TP	FN
Anomaly	FP	TN

The simple and often used performance indicators have been presented for IDS in the literature [38, 39]: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), detection rate and false alarm rate. In the presented work the following performance indicators are used for comparing the efficiency of classification algorithms:

##### **True positives rate**

$$TPR = \frac{TP}{positive}$$

TP represents the normal behavior which is correctly predicted as normal.

##### **FP- False positives rate**

$$FPR = \frac{FP}{(FP + TN)}$$

FP means that the anomalous behavior is predicted as normal

##### **True negatives rate**

$$TNR = \frac{TN}{negative}$$

TN means the anomaly behavior which is detected correctly.

<sup>5</sup> [https://rasbt.github.io/mlxtend/user\\_guide/classifier/StackingClassifier/](https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/).



**False negatives rate**

$$FNR = \frac{FN}{(FN + TP)}$$

FN - shows that the normal behavior which is wrongly thought as anomalous behavior.

We can also calculate precision, recall, F - measure, accuracy assessments after knowing the values of the above mentioned four parameters.

**Precision** is the measured proportion of the number of correctly predicted positive observations to the number of total positive observations.

$$precision = \frac{TP}{(TP + FP)}$$

**The recall** is a proportion of the number of correctly predicted positive observations to the number of total observations that their real class is yes.

$$recall = \frac{TP}{(TP + FN)}$$

**F - measure:** is expressed by the weighted average of precision and recall prices. □

$$F - measure = \frac{2 * (precision * recall)}{(precision + recall)}$$

**Accuracy** is the percentage of test set samples that are correctly classified by the model. In other words, this accuracy is expressed by the quotient of the number of correctly predicted observations to the number of total observations. Here correctly predicted observations may be from yes and no classes. Therefore TP and TN sum up for finding correctly predicted observations. And the number of total observations is expressed by the sum of the above-mentioned parameters. Then, the following formula is correct for accuracy [9].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

In the article anomaly detection model has been tested on the J48, LogitBoost, IBk, AdaBoost, RandomTree classifiers based on all indications vectors. As mentioned above, evaluating the effectiveness of the proposed model was conducted on the NSL-KDD open databases in the WEKA conditions. The **Stacking** method was used to create a classifier ensemble. Its general scheme is as follows.

**Scheme:** weka.classifiers.meta.Stacking -X 10 -M "weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\\\Program Files\\\\Weka-3-8\\" -seed 1" -S 1 -num-slots 1 -B "weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump" -B "weka.classifiers.meta.LogitBoost -P 100 -L -1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1 -E 1 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump" -B "weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last" -B "weka.classifiers.trees.J48 -C 0.25 -M 2" -B "weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1"

The SVM Radial Basis Function was taken as a meta classifier in the classifier ensemble.

Classifiers' detection precision was estimated based on precision, recall, false positive rate (FPR), true positive rate (TP), f-measurement (f-measure), accuracy metrics.

The result can be interpreted as follows. As can be seen in Table 3, the detection accuracy of the proposed approach for all metrics is higher than other methods. The accuracy of the anomalies of the proposed model is over 83 percent.

Table-3. Comparison of the Accuracy of Classifiers.

Methods	TP	FP	Precision	Recall	F- measure	Accuracy
Ada Bust	78.4%	17.4%	83.4%	78.4%	78.3%	78.44%
Random Tree	81.4%	16.0%	83.7%	81.4%	81.4%	81.36%
LBk	79.4%	16.5%	84.1%	79.4%	79.2%	79.36%
J48	81.5%	14.6%	85.8%	81.5%	81.5%	81.53%
Logit Boost	74.7%	21.0%	79.7%	74.7%	74.5%	74.72%
Proposed model	82.9%	12.0%	87.5%	82.9%	83.2%	<b>83.09%</b>

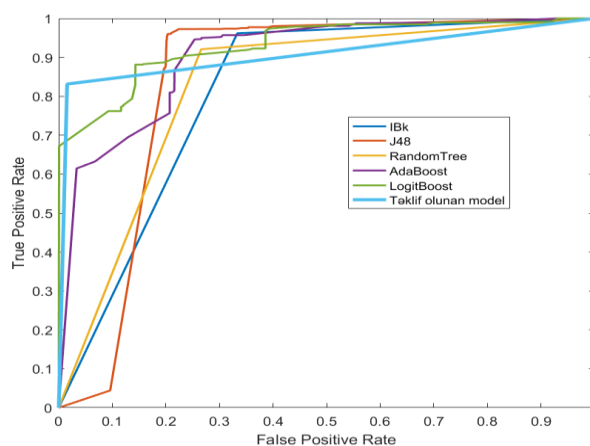


Figure-2. ROC curve.

FPR errors are low when the precision indicator is high, and FNR errors are low when the recall indicator is high. So it is difficult for both precision and recall indicators to be high at the same time. Because, in most cases when the recall indicator increases, the precision indicator is low and vice versa. F – measure shows that the model is exact(accurate) or not. This indicator shows that precision and recall indicators are high enough.

The ROC (Receiver operating characteristic) curve presented in Figure 2 was used as a visualization tool. For this purpose, Matlab software was used. In the curve, the x-axis reflects lies, and the y-axis reflects negative lies.

## 5. CONCLUSION

Network traffic anomalies can create serious disturbances for network security. The results of the network traffic analysis can help network engineers and administrators to create a more reliable network, avoid possible discharges and take precautionary measures. For this, there is a need to develop more effective methods of intellectual analysis. Within this work, the analysis of researchers shows that currently there are numerous approaches to the detection of anomalies. Proposed methods in various research differ with their positive and negative features. The specific features, conditions, factors, development tempo of every branch require the development of new methods and tools of decision-making support. The results of the presented multi-classifier model experiments allow the model to be applied to security objects. In the future, reducing the symptoms, optimization and so on. approaches for increasing of detecting accuracy of this model should be developed.

**Funding:** This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan – Grant № EIF-KETPL-2-2015-1(25)-56/05/1.

**Competing Interests:** The authors declare that they have no competing interests.

**Contributors/Acknowledgement:** Both authors contributed equally to the conception and design of the study.

## REFERENCES

- [1] M. S. Hajirahimova, "Big data technologies and information security challenges," *Problems of Information Technology*, vol. 1, pp. 49–56, 2016.
- [2] Y. Tian, "Towards the development of best data security for big data," *Communications and Network*, vol. 9, pp. 291-301, 2017. Available at: <https://doi.org/10.4236/cn.2017.94020>.

- [3] PwC's Global State of Information Security® Survey (GSISS), "Singapore highlights," Available: <https://www.pwc.com/sg/global-state-of-information-security-survey-2017-sg.pdf>, 2017.
- [4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey " *ACM Computing Surveys*, vol. 41, pp. 1-72, 2009.
- [5] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proceedings of the ACM Int. Conf. on Management of Data*, Santa Barbara, California, USA, 2001, pp. 37-46.
- [6] R. Zuech, "Intrusion detection and big heterogeneous data: A survey," *Journal of Big Data*, vol. 2, pp. 41-49, 2015.
- [7] M. Bai, X. Wang, J. Xin, and G. Wang, "An efficient algorithm for distributed density-based outlier detection on big data," *Neurocomputing*, vol. 181, pp. 19-28, 2016.
- [8] H. Kim, J. Kim, I. Kim, and T. M. Chung, "Behavior-based anomaly detection on big data," in *Proceedings of the 13th Australian Information Security Management Conference, 30 November – 2 December, Edith Cowan University*, Western Australia, 2015, pp. 73-80.
- [9] O. Tene and J. Polonetsky, "Big data for all: Privacy and user control in the age of analytics," *Northwestern Journal of Technology and Intellectual Property*, vol. 11, pp. 239-273, 2013.
- [10] R. M. Alguliyev, R. M. Aliguliyev, Y. N. Imamverdiyev, and L. V. Sukhostat, "An anomaly detection based on optimization," *International Journal of Intelligent Systems and Applications*, vol. 12, pp. 87-96, 2017.
- [11] R. Alguliyev and M. Hajirahimova, "Big data phenomenon: Challenges and opportunities," *Problems of Information Technology*, vol. 2, pp. 3-16, 2014.
- [12] S. Akbar, K. N. Rao, and J. Chandulal, "Intrusion detection system methodologies based on data analysis," *International Journal of Computer Applications*, vol. 5, pp. 10-20, 2010. Available at: <https://doi.org/10.5120/892-1266>.
- [13] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85-126, 2004. Available at: <https://doi.org/10.1007/s10462-004-4304-y>.
- [14] S. Dua and X. Du, *Data mining and machine learning in cybersecurity* vol. 256. USA: CRC Press, 2011.
- [15] J. Wang, D. Rossell, C. G. Cassandras, and I. C. Paschalidis, "Network anomaly detection: A survey and comparative analysis of stochastic and deterministic methods," presented at the 52nd IEEE Conference on Decision and Control). IEEE, 2013.
- [16] W. Wang, D. Lu, X. Zhou, B. Zhang, and J. Mu, "Statistical wavelet-based anomaly detection in big data with compressive sensing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, pp. 269-281, 2013. Available at: <https://doi.org/10.1186/1687-1499-2013-269>.
- [17] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 1153-1176, 2016. Available at: <https://doi.org/10.1109/comst.2015.2494502>.
- [18] S. Adibi, "Traffic classification—packet-, flow-, and application-based approaches," *International Journal of Advanced Computer Science and Applications*, vol. 1, pp. 6-15, 2010. Available at: <https://doi.org/10.14569/ijacsa.2010.010102>.
- [19] H. H. Pajouh, G. Dastghaibifard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *Journal of Intelligent Information Systems*, vol. 48, pp. 61-74, 2017. Available at: <https://doi.org/10.1007/s10844-015-0388-x>.
- [20] R. M. Alguliyev, R. M. Aliguliyev, and F. J. Abdullayeva, "Hybridisation of classifiers for anomaly detection in big data," *International Journal of Big Data Intelligence*, vol. 6, pp. 11-19, 2019. Available at: <https://doi.org/10.1504/ijbdi.2019.10018528>.
- [21] S. Aljawarneh, M. Aldwairi, and M. B. Yasin, "Anomaly-based intrusion detection system through feature selection analysis and building a hybrid efficient model," *Science Journal of Computational*, vol. 25, pp. 152-160, 2018. Available at: <https://doi.org/10.1016/j.jocs.2017.03.006>.

- [22] A. Branitskiy and I. Kotenko, "Hybridization of computational intelligence methods for attack detection in computer networks," *Journal of Computational Science*, vol. 23, pp. 145-156, 2017. Available at: <https://doi.org/10.1016/j.jocs.2016.07.010>.
- [23] W. Lee, R. A. Nimbalkar, K. K. Yee, S. B. Patil, P. H. Desai, T. T. Tran, and S. J. Stolfo, "A data mining and CIDF based approach for detecting novel and distributed intrusions," in *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection, October 02 - 04, 2000*, Springer-Verlag London, 2000, pp. 49 - 65.
- [24] Y. Imamverdiyev and L. Sukhostat, "Network traffic anomalies detection based on informative features," *Radio Electronics, Computer Science, Control*, pp. 113-120, 2017. Available at: <https://doi.org/10.15588/1607-3274-2017-3-13>.
- [25] M. Reháč, P. Michal, G. Martin, and B. Karel, "Trust-based classifier combination for network anomaly detection," in *Proceedings of the 12th International Workshop on Cooperative Information Agents XII, Springer-Verlag*, 2008, pp. 116-130.
- [26] M. Crosbie and E. H. Spafford, "Active defense of a computer system using autonomous agents," Technical Report CSD-TR- 95-008, Purdue Univ., West Lafayette, IN, 15 February 1995.
- [27] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2191-2204, 2003.
- [28] A. Mohiuddin, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016. Available at: <https://doi.org/10.1016/j.jnca.2015.11.016>.
- [29] KDD CUP, "KDD CUP 99 intrusion detection datasets." Available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>, 1999.
- [30] NSL-KDD, "NSL-KDD dataset for network-based intrusion detection systems." Available: <http://nsl.cs.unb.ca/NSL-KDD/>, 2017.
- [31] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, pp. 446-452, 2015.
- [32] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology*, vol. 2, pp. 1848-1853, 2013.
- [33] C. C. Aggarwal, *Data classification: Algorithms and applications*: Chapman and Hall/CRC Data Mining and Knowledge Discovery Series. Available: <https://www.crcpress.com/Data-Classification-Algorithms-and-Applications/Aggarwal/p/book/9781466586741>, 2014.
- [34] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *Springer Plus*, vol. 5, pp. 1304-1316, 2016. Available at: <https://doi.org/10.1186/s40064-016-2941-7>.
- [35] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119-139, 1997.
- [36] N. Didrik, "Tree boosting with XGboost - why does XGboost win "every" machine learning competition?," Master Thesis, 2016.
- [37] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241-259, 1992.
- [38] T. Holz, "Security measurements and metrics for networks," *Lecture Notes in Computer Science*, vol. 4909, pp. 157-165, 2008.
- [39] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skori, "Measuring intrusion detection capability: An information-theoretic approach," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, 2006, pp. 90-101.

*Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.*