Babak R. Nabiyev
Institute of Information Technology of ANAS, Baku, Azerbaijan
babek@iit.ab.az
DOI: 10.25045/jpit.v09.i1.11

APPLICATION OF CLUSTERING METHODS NETWORK TRAFFIC FOR DETECTING DDOS ATTACKS

One of the important problems of network security is availability. One of the most common threats to the network access are DDoS attacks. Identifying and preventing these attacks is the main purpose of this article. For this purpose, the data and methods of the KDD CUP 99 cluster were selected for their analysis. As the main methods of analysis, algorithms were chosen k-means and EM.
Keywords: *DDoS, clustering, k-means, EM-algorithm, network traffic, kdd cup 99.*

Introduction

Information security includes confidentiality, accessibility and completeness. Threats to any of them can pose a risk to the stored information. DDoS (Distributed Denial of Service) and DoS (Denial of Service) force the availability. DoS / DDoS is a cyber-crime tool. This type of attack is realized through making any service or network unavailable, weakening the throughput of the network, or realized by applying more than the processing capabilities of any service. Thus, availability is failed in all possible ways. However, this is not always caused by an attack. In some cases, it can be experienced in poorly configured systems and networks. For example,denial of service may occur when a large number of people access to a news website at the same time and these accesses are not optimized properly.

Recent DDoS attacks do not require specific knowledge in the field of information technology. Thus, this attack can be ordered at a low price or realized by using special tools developed by the malefactors. Therefore, it is important to be constantly aware of the latest trends, tools and threats to ensure security in this area.

Neustar's "Worldwide DDoS Attacks and Cyber Insights Research Report 2017" [1] represents the scale of the events that have occurred. According to the report, 45% of "volumetric" attacks has 10 Gbit/sec. capacity, and 15% - 50 Gbit/sec. This is about twice as much as the corresponding figures for 2016. The number of attacks to corporate networks has increased by 15% compared to previous year. As a result of DDoS attacks, 43% of organizations lose approximately 250000 USD within an hour whereas the attack prevention takes at least 3 hours in 51% of cases. The most interesting fact is that 99% of organizations have tools to prevent DDoS attacks.

Obviously, despite the use of various tools, software, and equipment for preventing DDoS attacks, combating this threat is a very difficult and often time-consuming process.

To solve this problem, many organizations have conducted research. One of them is the *DARPA (Defense Advanced Research Projects Agency),* which operates under the US Department of Defense. *DARPA 1998 IDS (Intrusion Detection System)* evaluation program has been developed and managed by MIT Lincoln Laboratory. The goal is to evaluate the studies in the field of intervention detection. This article uses KDD CUP 99 data set, which is developed by DARPA, for the evaluation of the proposed method.

K-means and EM- clustering algorithms are used for data analysis. The data set contains two types of traffic, which are DDoS traffic and normal traffic. Evidently, in most cases, normal traffic is accidentally prevented either by administrator or any tools when preventing the DDoS traffic. Alternatively, DDoS traffic may access the network being recognized as normal traffic. Therefore, the main goal of this article is to obtain more accurate results and, in general, to prevent DDoS attacks.

Related studies

Data mining methods distinguish normal and anomaly traffic efficiently and with high precision [2, 3]. Information about network traffic is collected primarily from routers and server and analyzed in IDS systems for DDoS attacks detection.

For example, [4] suggests a K-means method approach for detection and prevention of DDoS attacks. "*CAIDA USCD, DDoS Attack 2007 dataset"* DDoS attack base is used for trials, while*CAIDA Anonymized Internet Traces 2008*-for the survey of normal traffic, and data mining method described above is applied.

[5] proposesclustering for the proactive detection of DDoS attacks, i.e. detecting attacks and determining their causes. It suggests a unique architecture for proactive detection of DDoS attacks, which includes variables as processor and agent, contact and compromise, and attacks. The procedures of DDoS attacks are reviewed here and variables are selected based on these features. *"2000 DARPA Intrusion Detection Scenario Specific Data Set"*is used to verify the proposed method.

Recently, the malefactors mainly apply the application level of OSI model for the implementation of DDoS attacks. In this case, the methods used for the network and transport layers of OSI model fail in preventing DDoS attacks. At the application level, web services are posed to risks most. Two-level analysis is applied for detecting DDoS attacks at the application level [6]. In other words, the behavior of users is examinedbased on weblogs and the difference between the DDoS attack detection system and the analysis performed at the application level is studied. *Sparse vector decomposition and rhythm matching (SVD-RM)* based on L-Kmeans method is proposed for the implementation of the clustering process.

Adaptive Clusterization method with ranking functions is proposed for detection of DDoS attacks [7]. First, the primary variables are selected based on network traffic analysis. To identify the cluster structure of the targeted data, the modified "Global K-means" algorithm is used as anincremental clustering algorithm base. Then the linear correlation coefficient is used to rank the properties. Finally, the results of the properties rankingare used to recalculate clusters.

HTTP-GET attacks targeted to the HTTP protocol is one of the distributed DoS attacks. Malefactors achieve the denial of service by sending massive requests to the Web server through this attack. [8] offers a new method to deal with these types of attacks. Thus, entropy based clustering method is proposed using the Bayes factors to determine the difference based on normal and anomalous traffic.

Although many methods have been developed so far to detect DDoS attacks, they have two common problems. They include the capacity to study DDoS intrusion detection system and to handle large volume of non-structured data. For the problem solutionit is required to develop aDDoS intrusion detection system capable of examining, adapting to new threats, and maintaining and processing large volumes of non-structured data. The most promising approach to this problem is shown in [9]. DDoS attacks detection based on neural network and realized HBase system and the Apache Hadoop cluster is proposed here.

KDD CUP 99 data collection

DARPA 1998 MAS assessment program was developed and managed by MIT Lincoln Laboratory. The goal is to evaluate the studies in the field of intrusion detection. These data have been collected for 9 weeks in an imitated network. For the imitation, the US Air Force computer network is taken as a basis. The data collection, as shown in [10], consists of 4 parts and includes 41 features (Table 1), which means up to 5 million traffic packages and 4Gb volumes.

Table 1

Signs of the KDD data set

No	Indication name	Definition
1	Duration	length (number of seconds) of the connection
2	Protocol_type	type of the protocol, e.g. tcp, udp, etc.
3	Service	network service on the destination e.g. http, telnet, etc.
4	Src_bytes	number of data bytes from source to destination
5	Dst_bytes	number of data bytes from destination to source
6	Flag	normal or error status of the connection
7	Land	1 if connection is from/to the same host/port; 0 otherwise
8	Wrong_fragment	number of ``wrong'' fragments
9	Urgent	number of urgent packets
10	Hot	number of ``hot'' indicators
11	#_failed_logins	number of failed login attempts
12	Logged_in	1 if successfully logged in; 0 otherwise
13	#_compromised	number of ``compromised'' conditions
14	Root_shell	1 if root shell is obtained; 0 otherwise
15	Su_attempted	1 if ``su root'' command attempted; 0 otherwise
16	#_root	number of ``root'' accesses
17	#_file_creations	number of file creation operations
18	#_shells	number of shell prompts
19	#_access_files	number of operations on access control files
20	#_outbound_cmds	number of outbound commands in an ftp session
21	Is_host_login	1 if the login belongs to the ``host'' list; 0 otherwise
22	Is_guest_login	1 if the login is a ``guest''login; 0 otherwise
23	Count	number of connections to the same host as the current connection in the past two seconds
24	Srv_count	number of connections to the same service as the current connection in the past two seconds
25	serror_rate	% of connections that have ``SYN'' errors
26	srv_serror_rate	% of connections that have ``SYN'' errors
27	rerror_rate	% of connections that have ``REJ'' errors
28	Srv_rerror_rate	% of connections that have ``REJ'' errors
29	Diff_srv_rate	% of connections to different services
30	Srv_rerror_rate	% of connections that have ``REJ'' errors
31	Srv_diff_host_rate	% of connections to different hosts
32	Dst_host_count	count of connections having the same destination host
33	Dst_host_srv_count	count of connections having the same destination host and using the same service
34	Dst_host_same_srv_rate	% of connections having the same destination host and using the same service
35	Dst_host_diff_srv_rate	% of different services on the current host

36	Dst_host_same_src_ port_rate	% of connections to the current host having the same src_port
37	Dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts
38	Dst_host_serror_rate	% of connections to the current host that have an S0 error
39	Dst_host_srv_serror_ rate	% of connections to the current host and specified service that have an S0error
40	Dst_host_rerror_rate	% of connections to the current host that have an RST error
41	Dst_host_srv_rerror_ rate	% of connections to the current host and specified service that have an RST error

Fundamental indications: The main indications are obtained from differential packages without taking into account the useful loading for transfer.

Content: In this case, the indications are used to evaluate useful loading for transfer and failed login attempts in TCP packets.

Time-based traffic indications: These functions are used to get the indications about the incidents occurring constantly over two seconds. For example, determining the number of connections to the host.

Host-based traffic indications: This indication applies not to the time, but the historical window to determine the number of connections. It is also used to determine the scale of attacks that occur over two seconds.

[11] provides the attack classes shown in the KDD CUP 99 data set as follows (Table 2):

1. DOS: denial-of-service, e.g. syn flood;
2. R2L: unauthorized access from a remote machine, e.g. guessing password;
3. U2R: unauthorized access to local super user (root) privileges, e.g., various ``buffer overflow'' attacks;
4. probing: surveillance and other probing, e.g., port scanning.

The KDD CUP 99 data collection has the following DoS attacks:

1) **backDoS**: an attack targeted at Apache server. Malefactor uses a large number of backslash (\) when applying to URL. The server's performance slows down while it is trying to process this request, and processing time of other requests extends or they are not processed at all. Consequently, denial of service occurs.

Table 2

The type, number, class, and indications of the historical windows in the KDD data set

Types of historical windows	**Number of historical windows**	**Classes of historical windows**	**Respective indications**
back	2,203	DoS	5,6
land	21	DoS	7
neptune	107,201	DoS	3.4,5,23,26.29,30,31,32,34,36,37,38,39
pod	264	DoS	8
smurf	280,790	DoS	2,3,5,6,12,25,29,30,32,36,37,39
teardrop	979	DoS	8
satan	1,589	PROBE	27
ipsweep	1,247	PROBE	36
nmap	231	PROBE	5
portsweep	1,040	PROBE	28
normal	97,277	NORMAL	3,6,12,23,25,26,29,30,33,34,35,36,37,38,39
Guess_passwd	53	R2L	11,6,3,4

ftp_write	8	R2L	9,23
imap	12	R2L	3,39
phf	4	R2L	6,10,14,5
multihop	7	R2L	23
warezmaster	20	R2L	6,1
warezclient	1,020	R2L	3,24,26
spy	2	R2L	39,1
Buffer_overflow	30	U2R	3,24,14,6
loadmodule	9	U2R	36,24,3
perl	3	U2R	14,16,18,5
rootkit	10	U2R	24,23,3

2) **landDoS**: malefactor can send a specially formatted packet to malfunction the servers located outside of the server, which ultimately leads to a service denial. This attack uses the TCP / IP protocol's vulnerability. For example, if the source IP address and port of the package targeted at the server are identical to the destination IP address and port, it is a fake package. However, this attack forces the server to request to itself, which consequently results in the service denial.

3) **neptuneDoS**: generates a large number of incomplete TCP / IP sessions to be processed leading to network or server hardware failure.

4) **Ping of death (PoD) DoS**: As it is seen from the name, the attack sends ping packets. However, this is not an ordinary ping pack, but an abnormal ping package of 64,000 bytes. When the server or network device receives such a big abnormal ping packet, the performance can fail or re-load.

5) **smurfDoS**: When the ping is sent, the source IP address of the package is changed to the target IP address. The ping is sent from this ping simultaneously to different locations. Since all the packaged computers are forced to respond to the Ping package, they all respond to this packet. If the traffic flow is generated with sufficient power, the targeted computer, the source IP address of which is specified, is exposed to the denial of service.

6) **teardropDoS**: If the packet is large, it is divided into smaller parts according to the rule, marked and re-assembled on the receiver. If the hacker changes in the markup of the packet and if the receiving computer does not take any measures in this regard, it is posed to the risk.

Additionally, the KDD CUP 99 data set contains about 494020 rows of logs about the traffic. These logs comprise 22 types of attacks and 1 normal traffic record(Table 2). 6 out of them are about DoS attacks. Since this article reviews DoS attack detection, 6 types of log rows out of KDD data set related to DoS attack are selected, i.e., 391458 rows and 97277 rows associated with normal traffic (table 2). Remained ones are removed from KDD data set(table 3). The outcome of this editing process will allow to perform both fast and accurate analysis.

Table 3

KDD data set after sorting

Names of records	**Number**
back	2203
teardrop	979
neptune	107201
land	21
smurf	280790
pod	264
normal	97277

Then a DoS class is created by combining six DoS attack classes. Thus, 2 classes are involved in the clustering process (Table 4). These are the DOS class that combines DoS attacks and NORMAL class that incorporates normal traffic.

Table 4

DoS attack classes after combining

Names of records	**Number**
DoS	391458
Normal	97277

The process of analysis

The process of analysis is based on the *K*-means and EM algorithm. The analysis is based on both algorithms on the data presented in Table 4.

Initially, the k-means clustering algorithm is applied to detect DoS attacks [12]. The data set $X = \{x_1, \dots, x_n\}$ consists of traffic sessions n. And each traffic-session is described as the m-dimensional pointin Euclidean-space x_i: $x_i = (x_{i1}, \dots, x_{im})$, where $x_{ij}$is the weight of the j-th attribute of the i-th traffic session, $(i = 1, \dots, n;\ j = 1, \dots, m)$. The goal is to split up the traffic-sessions, i.e., data set $X = \{x_1, \dots, x_n\}$into K number of clusters:$C = (C_1, \dots, C_K)$. Assume that the following conditions are provided:

1) $C_p \neq \varnothing$for arbitrary p , i.e., there must be at least one point in each cluster
2) $C_{p1} \bigcap C_{p2} = \varnothing$ for arbitrary $p1 \neq p2$, i.e., two different clusters should not have elements in common, $p1, p2 = 1,..., K$;
3) $\bigcup_{p=1}^{K} C_p = \mathrm{X}$, i.e., every point should be definitely assigned to any cluster;
4) There are no conditions for clusters C_p, $p = 1,..., K$.

The k-means algorithm consists of the following steps:

1. K number of points are first selected from the points set $X = \{x_1, \dots, x_n\}$as the center of clusters. These centers are denoted as $O = \{o_1, \dots, o_K\}$and $s = 0$ is accept (s indicates the number of iterations).
2. The distance between the center each$x_i = (x_{i1}, \dots, x_{im})$and the p-thcluster,$o_p = (o_{p1}, \dots, o_{pm})$ is calculated. The Euclidean metric is used to calculate this distance:

$$\mathrm{d}(x_i, o_p) = \left(\sum_{j=1}^{m} (x_{ij} - o_{pj})^2\right)^{\frac{1}{2}},\ i = 1, \dots, n;\ p = 1, \dots, K, \qquad (1)$$

where o_{pj}- is the j-th coordinate of the center of the p-th cluster.

3. The pointx_i refers to the cluster, where the value of $\mathrm{d}(x_i, o_p)$ is minimum, i.e., $x_i \in C_p$ if $d(x_i, o_p) = \min_q d(x_i, o_q)$.
4. After all the points are assigned to the clusters, the following goal is calculated:

$$f^{(s)}(x) = \sum_{p=1}^{\mathrm{K}} \sum_{\mathrm{x} \in C_p} \|x - o_p\|^2 \qquad (2)$$

The smaller the value of this function, the better clustering is.

5. Then the center of each cluster is recalculated with the following formula:

$$o_p = \frac{1}{|C_p|} \sum_{x_i \in C_p} x_i,\ p = 1,..., K, \qquad (3)$$

where $|C_p|$- is the number of points in the p -th cluster.

6. $s = s + 1$
7. Steps 3 to 5 are repeated if the following collection conditions are provided:

$$\left| \frac{f^{(s+1)}(x) - f^{(s)}(x)}{f^{(s)}(x)} \right| \le \varepsilon \tag{4}$$

where ε is the predefined parameter.

The following index is used to evaluate the clustering quality [13]:

$$\text{Validity} = \frac{\sum_{p=1}^{K} \left\{ \frac{1}{|C_p|} \max_{x_i \in C_p} \mathrm{d}(x_i, o_p) \right\}}{\sum_{p=1}^{K} \left\{ \min_{\substack{q \neq p \\ q=1,\ldots,K}} \mathrm{d}(o_p, o_q) \right\}} \tag{5}$$

The smaller the value of this index, the higher the cluster quality is.

The second method to be applied is the EM algorithm. Since the EM algorithm evaluates the parameters to maximize the probability of the monitored information. For this purpose, the information is given about the probability log $L_c(\Psi)$ repeated between two steps.

The step E of EM algorithm consists of the following computing:

$$Q(\Psi, \Psi^{(q)}) = E_{\Psi(q)} [\log L_c(\Psi) | y, z];$$

Here $\Psi^{(q)}$ corresponds to the iteration q of Ψ. Whereas, $E_{\Psi(q)}$ represents the mathematical expectation computed to use the parameters $\Psi_{(q)}$, i.e.

$$t_{ik} = E_{\Psi(q)}[Z_{ik} | x_i, \Psi_k] = P_{\Psi(q)}[Z_{ik} = 1 | x_i] = \frac{\pi_k g_k(x_i; \Psi_k)}{\sum_{k=1}^{g} \pi_k g_k(x_i; \Psi_k)}; \tag{6}$$

then

$$Q(\Psi, \Psi^{(q)}) = \sum_{k=1}^{g} log \pi_k \sum_{i=1}^{n} t_{ik} - \frac{np}{2} log(2\pi) - \sum_{k=1}^{g} \sum_{i=1}^{n} t_{ik} \sum_{j=1}^{p} \log(\sigma_{jk}) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{g} \sum_{j=1}^{p} \frac{t_{ik}}{\sigma_{jk}^2} (x_{ij} - m_{jk})^2$$

The step *M* of EM algorithm considers the maximization of the expectation $\Psi^{(q)}$ related to $Q(\Psi, \Psi^{(q)})$, i.e., it is defined as $Q(\Psi_{q+1}, \Psi^{(q)}) \ge Q(\Psi, \Psi^{(q)})$ when calculating Ψ_{q+1} and $\Psi \in \Omega$ is accepted for all. In practice, Ψ equals to zero for each component of derivatives $Q(\Psi, \Psi^{(q)})$ of the updating equations. If the covariance matrix is accepted diagonally, then

$$\pi_k^{(q+1)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}$$

$$m_{jk}^{(q+1)} = \frac{\sum_{i=1}^{n} t_{ik} x_{ij}}{\sum_{i=1}^{n} t_{ik}}$$

$$\sigma_{jk}^{(q+1)} = \sqrt{\frac{\sum_{i=1}^{n} t_{ik} \left(x_{ij} - m_{jk}^{(q+1)} \right)^2}{\sum_{i=1}^{n} t_{ik}}}$$

The clustering process is based on the above-mentioned and modified KDD data set and on the Expectation-maximization (EM) clustering algorithm located on the WEKA tool [14, 15]. EM clustering algorithm is similar to K-means method. Basic operations of K-means clustering are relatively simple. Given the defined number of clusters *K*, these clusters are controlled to ensure the clusters for all variables to be as far as possible from one another. The EM algorithm expands this base approach in two important ways:

- Instead of specifying samples for clusters and maximizing the difference between continuous variables, the EM cluster algorithm calculates the probability of a cluster membership by one or more probability distributions. The main purpose of the clustering algorithm is to maximize the probability of data by taking into account the common probability or clusters.
- Unlike the classic k-means clustering realization, EM algorithm can be applied to both uninterrupted and category variables.

WEKA tool is based on the Intel Xeon x5670 2-core 2.93Ghz processor and 12Gb memory-mounted computer on VMware virtual machine. In the course of the analysis, 10-step cross-check interval was selected. The results of clustering in this case are as follows (Table 5):

Table 5

Results of K-means method for two clusters

0 (DoS)	1(Normal)	< - -Defined clusters
280947	110511	DoS
1107	96170	Normal

The error pace of the result is 22.8381%. Obviously, this result is inexact and does not represent the reality. Then, EM algorithm was applied based on the same data and same conditions. In this case, clustering results are as follows (Table 6):

Table 6

Results of EM algorithm on two clusters

0 (DoS)	1 (Normal)	< - - Defined clusters
107359	284099	DoS
683	96594	Normal

The error pace of the result is 41.7308%. Apparently, these results are also inexact and do not represent the reality.

The reason for this error is that the characteristics of Smurf DoS traffic are similar to the characteristics of the normal traffic. Therefore, DoS class was divided into 2 parts in order to achieve more accurate and complete clustering results. DoS1 (*teardrop, neptune, land, pod*), DoS2 (smurf). In this case, we already possess 3 classes, which are DoS1, DoS2, and NORMAL classes. Accordingly, the parameters provided in the 1st experiment were re-applied to 3 classes. The results obtained are as follows for *K*-means method (Table 7):

Table 7

Results of K-means method on three clusters

0 (DoS2)	1 (Normal)	2 (DoS1)	< - - Defined clusters
159	3200	107309	DoS 1
280788	2	0	DoS 2
736	96518	23	Normal

In this case, the error rate of the result is 0.843.

Afterward, EM algorithm is applied for 3 classes and the error rate of the result equals to 0.8162. Distribution of obtained results by clusters is as follows (Table 8).

Table 8

Results of EM algorithm on three clusters

0 (DoS1)	1 (DoS2)	2 (Normal)	< - - Defined clusters
107175	0	3493	DoS1
0	280327	463	DoS2
33	0	97244	Normal

TP, TN, FP, FN metrics are used to evaluate the quality of clustering results. For this purpose, *Introduction to Information Retrieval* book by *The Stanford Natural Language Processing Group* uses a program proposed in [16] to solve three-cluster clustering measuring problem. The results are as follows (Table 9):

Table 9

Clustering quality metrics

Clustering metrics / Quality metrics	*K*-means	EM algorithm
TP	49841582762	49769015431
FP	562947427	389851298
TN	68591380159	68764476288
FN	434795397	507362728
Rand index	0.991646	0.992488
Precision	0.988831	0.992228
Recall	0.991352	0.989909
F1	0.990090	0.991067

Conclusion

This article described the process of detecting abnormal traffic and extracting from normal traffic based on clustering method. DDoS attacks and normal traffic were extracted from the data set DARPA KDD CUP 99. Experiments were conducted on selected data sets. EM clustering algorithm was applied to the data set analysis. At the first stage, 6 types of DoS attacks and normal traffic were divided into two classes and the EM algorithm was applied. However, the results were unsatisfactory. As a result of the study, it was concluded that this was due to the fact that the smurf DoS attack traffic characteristics were similar with the normal traffic characteristics. After dividing the DoS class into two parts, more precise clustering results were obtained.

This work was implemented with the financial support of the Science Development Fund under the President of the Republic of Azerbaijan - Grant No. EIF-KETPL-2-2015-1 (25) -56 / 05/1

References

1. https://www.neustar.biz/about-us/news-room/press-releases/2017/dDoS2017
2. Bhaya W., Manaa M.E. Review clustering mechanisms of distributed denial of service attacks // Journal of Computer Science, 2014, vol.10, no.10, pp.2037–2046.
3. Bhuyan M.H., Kashyap H.J., Bhattacharyya D.K., Kalita J.K. Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions // The Computer Journal, 2013, vol.57, no.4, pp.537–556.

4. Bhaya W., Manaa M.E. A Proactive DDoS Attack Detection Approach Using Data Mining Cluster Analysis // Journal of Next Generation Information Technology, 2014, vol.5, no.4, pp.36–47.
5. Lee K., Kim J., Kwon K. H., Han Y., Kim S. DDoS attack detection method using cluster analysis // Expert Systems with Applications, 2008, vol.34, no.3, pp.1659–1665.
6. Liao Q., Li H., Kang S., Liu C. Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching // Security and Communication Networks, 2015, vol.8 , no.17, pp.3111–3120.
7. Zi L., Yearwood J., Wu X.W. Adaptive Clustering with Feature Ranking for DDoS Attacks Detection / International Conference on Network and System Security (NSS), 2010, pp.281–286.
8. Chwalinski P., Belavkin R., Cheng X. Detection of Application Layer DDoS Attacks with Clustering and Bayes Factors / International Conference on Systems, Man, and Cybernetics, 2013, pp.156–161.
9. Zhao T., Lo D.C., Qian K. A Neural-Network Based DDoS Detection System Using Hadoop and HBase / 17th International Conference on High Performance Computing and Communications (HPCC), 2015, pp.1326–1331.
10. Kayacık H. G., Zincir-Heywood A. N., Heywood M.I. Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets / Third Annual Conference on Privacy, Security and Trust, 2005, pp.1–6.
11. Olusola A. A., Oladele A. S., Abosede D. O., Analysis of KDD'99 Intrusion Detection Dataset for Selection of Relevance Features / Proceedings of The World Congress on Engineering and Computer Science, 2010, pp.162–168.
12. Kumari R., Sheetanshu, Singh M.K., Jha R., Singh N.K. Anomaly detection in network traffic using K-mean clustering // International Conference on Recent Advances in Information Technology (RAIT), 2016, pp.387–393.
13. Aliguliyev R.M. Performance evaluation of density-based clustering methods // Information Sciences, 2009, vol.179, no.20, pp.3583–3602.
14. Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set // IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2009, pp.53–58.
15. Quost B., Denœux T. Clustering fuzzy data using the fuzzy EM algorithm // Fuzzy Sets and Systems, 2016, vol.286, pp.134–156.
16. http://stats.stackexchange.com/questions/89030/rand-index-calculation