A LIGHTWEIGHT METHOD FOR DETECTING CYBER ATTACKS IN HIGH-TRAFFIC LARGE NETWORKS BASED ON CLUSTERING TECHNIQUES

Nguyen Hong Son[1] and Ha Thanh Dung[2]

[1]Faculty of Information Technology
Posts and Telecommunications Institute of Technology, Vietnam
[2]Falculty of Information Technology
Saigon University, Vietnam

ABSTRACT

Protecting information systems is a difficult and long-term task. The size and traffic intensity of computer networks are diverse and no one protection solution is universal for all cases. A certain solution protects well in the campus network, but it is unlikely to protect well in the service provider's network. A key component of a cyber defence system is a network attack detector. This component needs to be designed to have a good way to scale detection capabilities with network size and traffic intensity beyond the size and intensity of a campus network. From this point of view, this paper aims to build a network attack detection method suitable for the scale of large and high-traffic networks based on machine learning models using clustering techniques and our proposed detection technique. The detection technique is different from outlier detection commonly used in clustering-based anomaly detection applications. The method was evaluated in cases using different feature extraction methods and different clustering algorithms. Experimental results on the NSL-KDD data set are positive with a detection accuracy of over 97%.

KEYWORDS

Cyberattack Detection System, Clustering Techniques, High-Traffic Networks, Cluster Feature Vector

1. INTRODUCTION

The effectiveness of defence systems against cyberattacks depends on the capabilities of the network attack detection component. In industrial networks, the detection component is built into Intrusion Detection Systems (IDS) or Intrusion Detection and Prevention System (IDP). Once an attack is detected, stopping it is not too difficult. The ability of the network attack detection component is reflected in the accurate warning when the attack takes place and the immediate warning when the first signs of the attack appear, often in short, early warning. However, the early warning and error-free capabilities of today's detection solutions are still to be desired. The reason is that attacks are diverse, and constantly changing and information system infrastructures also have their characteristics, making it difficult to track and detect attacks. Thereare no single IDS that can monitor and alert the entire information system, so depending on the scope of responsibility, IDS is divided into two types: Host Intrusion Detection System (HIDS) and Network Intrusion Detection System (NIDS). HIDSs can only detect attacks on end systems and NIDSs can only detect attacks on the network.

In a general perspective, regardless of the theory or technology used, attack detection methods fall into one of the following three main categories:

(i) Based on known attack behaviours, each attack is identified by a unique signature, also referred to as the signature-based detection method.
(ii) Based on known normal valid behaviours, also referred to as anomaly-based detection method.
(iii) Based on a predetermined threshold of a measurement parameter selected in the design of the method, also referred to as the statistical-based detection method.

In the methods using the form (i), the detection unit continuously monitors the activities on the information system and looks for signs that match the known attack signs, if any, it will emit a warning [1-4]. The effectiveness of these methods depends on the knowledge of the known attack signature and the processing power of the hardware infrastructure running the detection module. This method cannot detect unknown attacks. In methods using form (ii) the detector also continuously monitors and checks the activities taking place on the information system and issues an alert when there is an activity that is different from the normal known activities [5-7]. Thus, the effectiveness of methods of this type depends on knowledge of the normal operations and processing power of the hardware infrastructure running the detection module. This approach can detect unknown attacks but can still be mistaken without full knowledge of the normal behaviour. Methods using form (iii) creatively define anomaly measurement parameters on the information systems, when the value exceeds the specified threshold, it will issue an intrusion alarm [8-11]. The effectiveness of methods of this type depends on the reliability and suitability of the parameters established for various types of attacks and the threshold value chosen.

In recent years many attack detection methods use machine learning and deep learning techniques to improve the accuracy of the method [12-15]. The main job is to build an attack detection model according to two learning methods: supervised learning and unsupervised learning. Supervised learning requires a labelled data set to train the model. The trained model acts as a classifier between the normal data and the attack data, which is equivalent to the form (i) mentioned above. Meanwhile, attack detection models are built based on unsupervised learning methods using unlabeled data sets to train the model and the trained model acts as a cluster. Once the data is fed into the model, the normal data is distributed into clusters and the attack data becomes outliers that are the basis for attack detection. The way to build such an unsupervised learning-based detection model is equivalent to the form (ii) mentioned above. Thus, the attack detection model based on unsupervised learning can detect unknown attacks. This is very important in practice because it is not easy to have all the data labelled and new types of attacks are constantly emerging. However, the limitation of current unsupervised models to detect attacks based on outliers is low accuracy [16-17]. Attack detection models using supervised or unsupervised learning both operate on the principle of data point detection, which has two shortcomings: The first shortcoming is that the manifestation of an attack is not only contained in a certain data point but can include many data points. Therefore, efforts to use classification methods are difficult to achieve high accuracy in the case of complex attacks. The second shortcoming is the fact that live streaming data goes into the model and when the traffic increases, the models are very difficult to handle.

In this paper, we propose an attack detection method in the form (i) but using clustering techniques can overcome the above limitations. As a result, the model can be applied appropriately to high-traffic infrastructures. The main contributions of the paper include:

- The method of determining the full manifestation of an attack is based on clustering techniques, whereby actual attacks that take place in complex steps can be controlled.
- The attack detection method can be implemented in a distributed parallelism model suitable for large networks with high traffic.

Here we do not create a classifier to test each data point for attack or not but create a cluster, which processes the data in batches depending on the sampling period to detect the attack. Model training is also not about isolating data into clusters and separating attack data points into outliers. Instead, we cluster all the data points and accurately determine the characteristic expression of the anomalous clusters as a basis for detection according to the form (i) mentioned above. We call the characteristic manifestations of the cluster the cluster feature vector. Based on batch data processing, detected attack activities based on cluster feature vectors are matched with known anomalous feature vectors in the trained model.

To increase the accuracy of clustering, we preprocess the data and use the appropriate feature extraction method. In this paper, we will use two different feature extraction methods and different clustering algorithms, in turn, to see how the performance of different cases is. Specifically, we apply Risk-based Acquisition [18] and Attribute Ratio [19] feature extraction methods, respectively, and also use two clustering algorithms K-means and DBSCAN respectively. Experiments were conducted with the NSL-KDD dataset [20]. The authors in [20] argue that the size of the NSL-KDD dataset is reasonable, and can be used as a complete data set without the need for random sampling.

The rest of the paper is organized as follows: Section 2 presents some typical attack detection methods in recent years that have similarities with one aspect of our method. Section 3 summarizes the theoretical topics used in the proposed method. Details of the proposed detection method and model building are presented in Section 4. Section 5 presents method testing and evaluation, including use cases of different preprocessing methods and clustering algorithms. The paper ends with the conclusions in Section 6.

2. RELATED WORKS

Intrusion detection systems (IDS) are developed based on methods of distinguishing normal and abnormal activities on computer networks. Many differentiating methods have been introduced and applied in practice, each with its advantages and disadvantages. Which, the group of methods using clustering techniques has also attracted the attention of many researchers [21]. Recently, the authors in [22] have proposed an intrusion detection method that combines the K-means clustering algorithm and Isolation Forest. This method is also intended for attack detection in big data systems in industrial environments. Thereby also shows that using the clustering technique creates favourable conditions for implementing detection solutions in large-scale data cases. Experimental results on the KDD 99 dataset achieved an AUC of 0.96 and an AUC of 0.98 on the Breast dataset. Regarding the security of the cloud computing system, [23] also proposes a solution to detect DDoS attacks on cloud computing based on network data clustering. The authors in [23] used the PCA algorithm for feature extraction to increase the efficiency of clustering. Experimental cases applied with K-means, DBSCAN, and Agglomerative algorithms are also evaluated, the Adjusted Rand Index metric is above 0.8989 and other metrics are also positive.

Anomaly detection based on clustering techniques is also applied in error detection of the system, as the authors in [24] have proposed a solution for anomaly detection on machine tools. Which, the Mean Shift clustering algorithm is used to identify repeating patterns in combination with the self-organizing map to provide information about the machine state to help detect anomalies with high efficiency.

Anomaly detection methods should all be based on a deep understanding of the monitored data. How the data features are exploited depends on the design of the method. In [25] shows that each type of attack has a different set of important features. On that basis, if the feature extraction is

correct, the classifier will work very well. Also related to detection based on its own set of features of each type of attack, in [26-27] the authors propose a DDoS low-rate attack detection method. This method will let the system learn to represent knowledge about low-rate DDoS attacks in the form of a set of feature vectors, labeling these feature vectors corresponding to the types of low-rate DDoS attacks. Feature vectors are built based on the botnet's features and the self-similarity of the traffic. In the detection stage, the semi-supervised fuzzy c-mean clustering algorithm is applied and assigns a feature vector to each cluster. As a result, low-rate DDoS can be detected with an accuracy of 97.46%.

3. BACKGROUNDS

3.1. K-Means Algorithm

K-Means clustering algorithm is proposed in [28]. K-Means is a commonly used algorithm in data clustering applications. The main idea of the K-Means algorithm is to find away to group the objects in a given data set into k clusters $\{C_1, C_2, \ldots, C_k\}$. The data set consisting of n objects in d-dimensional space $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ (i = 1… n)is clustered so that the standard function $E = \sum_{i=1}^{k} \sum_{x \in Ci} D^2 (x - m_i)$ reaches the minimum value, where: m_i is the centroid of the cluster C_i and D is the distance between the two objects.

3.2. DBSCAN Algorithm

DBSCAN clustering algorithm was introduced in [29] when the authors studied spatial data clustering algorithms based on the definition of a cluster as the maximum set of connected points in terms of density. The main idea of DBSCAN-based detection is that there is always a higher density inside each cluster than outside the cluster. Furthermore, the density in noisy regions is lower than the inner density of any cluster. DBSCAN uses Eps and MinPts parameters in the algorithm to control the density of clusters. Each cluster must determine the neighborhood radius (Eps) and the minimum number of points in the vicinity of a point in the cluster (MinPts). The neighbourhood of a point is determined based on the distance function between two points p and q, denoted dist(p,q).

3.3. Risk-based Acquisition method

Risk-based Acquisition is a feature extraction method proposed in [18]. The authors have shown how to calculate the risk value for the service attribute and the flag attribute in the network attack dataset. The risk value is calculated by the formula (1)

$$W_i = 1 - P(\text{normal}|x_i) \; \forall i = 1,\ldots,C \qquad (1)$$

Where W_i is the risk value, normal is the normal connection type (normal) and C is the number of attack types.

Replacing the corresponding risk values with each value of the service attribute and the flag attribute effectively improved the F-Score and other metrics, and reduced runtime when compared to the One Hot Encoding method.

3.4. Attribute Ratio Method

Attribute Ratio is a feature extraction method proposed in [19]. The authors use the average value of numeric attributes and the frequency of occurrence of binary attributes corresponding to each type of attack in the data set. Attribute Ratio is calculated by the formula (2):

$$AR(i)=MAX(CR(j)) \quad (2)$$

Where CR is a scale attribute of the class representing the ith attribute.
CR is calculated using two expressions corresponding to each attribute type. For attributes of numeric type, CR is calculated by expression (3):

$$CR(j) = \frac{AVG(C(j))}{AVG(total)} \quad (3)$$

For an attribute of binary type CR calculated using expression (4):

$$CR(j) = \frac{Frequency(1)}{Frequency(0)} \quad (4)$$

4. PROPOSED DETECTION METHOD

4.1. Working Principle of the Proposed Network Attack Detection System

Our proposed network attack detection application has the operating principle described in Figure 1. Traffic from protected network partitions is continuously collected and stored on temporary storage. The network traffic data is kept in its raw form and is passed batch by batch by the loader into a trained model for detection. At the input of the model, the raw data batch is preprocessed and features are extracted. Next, the data with the extracted features are fed into the clustering algorithm. All clusters formed at the output of the clustering algorithm will be calculated to determine the cluster feature vector and checked by comparing it with the vectors in the set of known attack feature vectors in the trained model. If the feature vector of a new cluster is similar to an anomalous feature vector, the notifier will issue an attack alert.

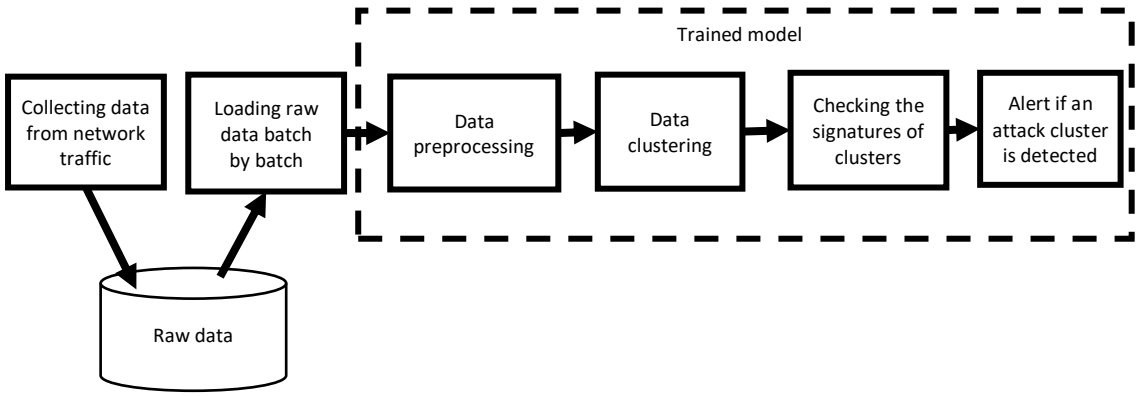

Figure 1. Overview of operation of the proposed cyberattack detection system

Our proposed detection system can fully apply parallelism to increase speed and thus be able to accommodate high-traffic infrastructure. The architecture of the detection system that allows parallel processing is shown in Figure 2. First of all, input batch data loading and processing can

be done in parallel by running multiple detection processes at the same time, each responsible for different batches, batch i different from batch j, as shown in Figure 2. There are many clusters formed after the clustering process and need to check each cluster to detect. This is great for opening multiple cluster test processes running in parallel, like worker 1, worker 2, worker 3, and worker 4 in Figure 2. Each worker checks a different number of clusters, a total of n and m clusters, one worker checks n clusters, and the other worker checks m clusters. The number n may or may not be equal to m depending on the load generated by the clusters and the worker's capacity. The number of attack detection processes as well as the number of parallel workers can scale depending on how much traffic needs to be handled on the high-traffic network infrastructure.

4.2. The Proposed Attack Detection Model

4.2.1. The Process of Developing the Model

Model building work is carried out through model design, model training, and model evaluation. The architecture of the model is shown in Figure 3, including the preprocessor, clustering unit, cluster signature computation unit, and attack cluster signature storage unit.

Regarding model training, we do not aim to build the model as a classifier like conventional supervised learning models. We also do not rely on unsupervised clustering techniques to detect anomalies based on outliers. Instead, we use clustering techniques to isolate attack data into clusters and find their signature. To do this, we conduct the training process as shown in Figure 3. The input training dataset is fed to the preprocessor and feature extraction, using one of two methods: Risk-based Acquisition (RA) and Attribute Ratio (AR). All pre-processed data is fed into the clustering algorithm to convert into clusters, here we use one of two clustering algorithms in turn: K-means and DBSCAN. Next, all clusters go through a computational process to determine the characteristic signatures of each cluster. The final result of the training process is a model with a stored set of signatures of attack clusters. We call the signatures the attack cluster feature vectors. Thus, the set of attack cluster feature vectors is also the result of the training process. The model is trained according to Algorithm 1.

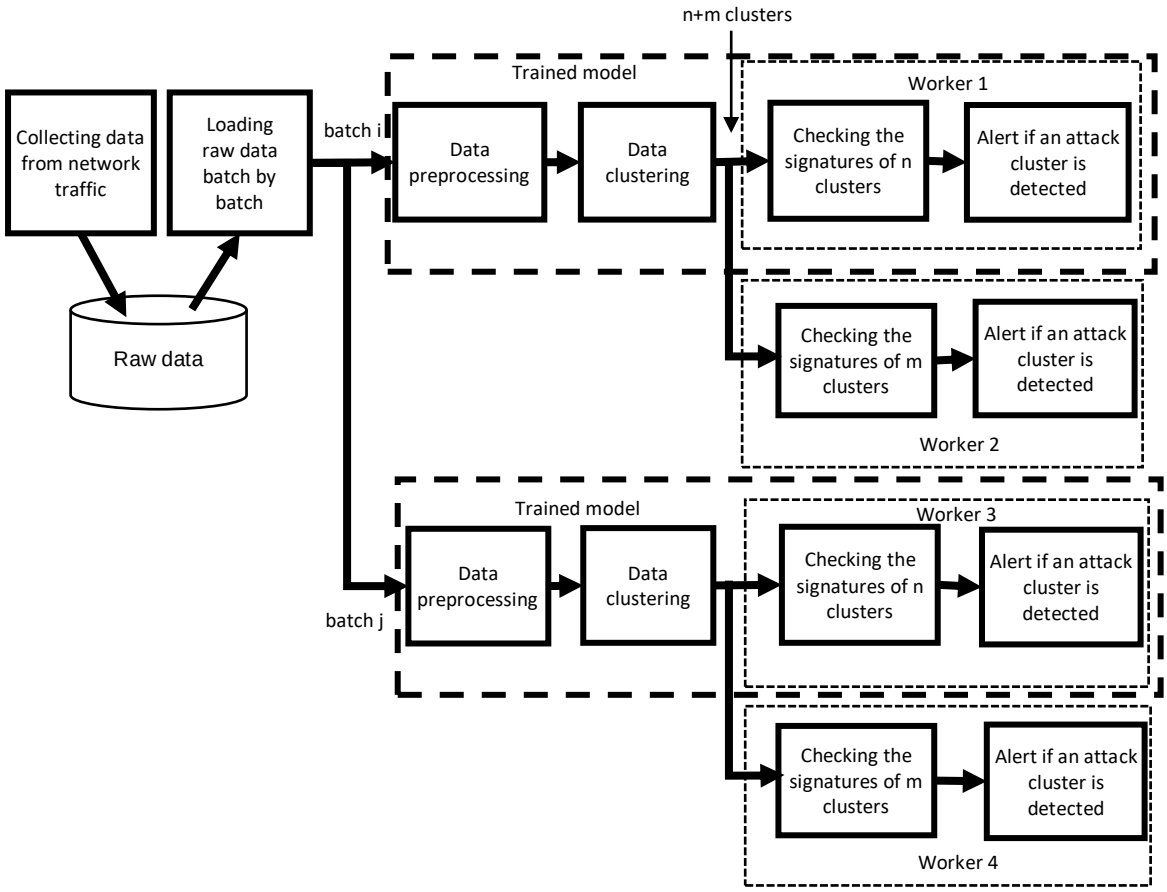

Figure 2. Parallel-enabled cyberattack detection system.

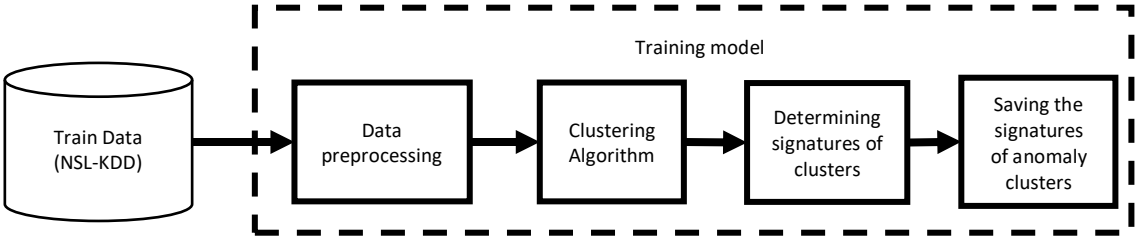

Figure 3. The model training process.

Attack detection is done according to Algorithm 2. The model is evaluated against the test data set following the steps described in Figure 4. The test data set is entered into the trained model, and the output of the model is the attack prediction result. The results will be matched with the ground truth in the test dataset to evaluate. The evaluation uses a confusion matrix with parameters ACC (Accuracy), True Positive Rate (TPR), False Positive Rate (FTP), PR (Precision Rate), and F1 Score.

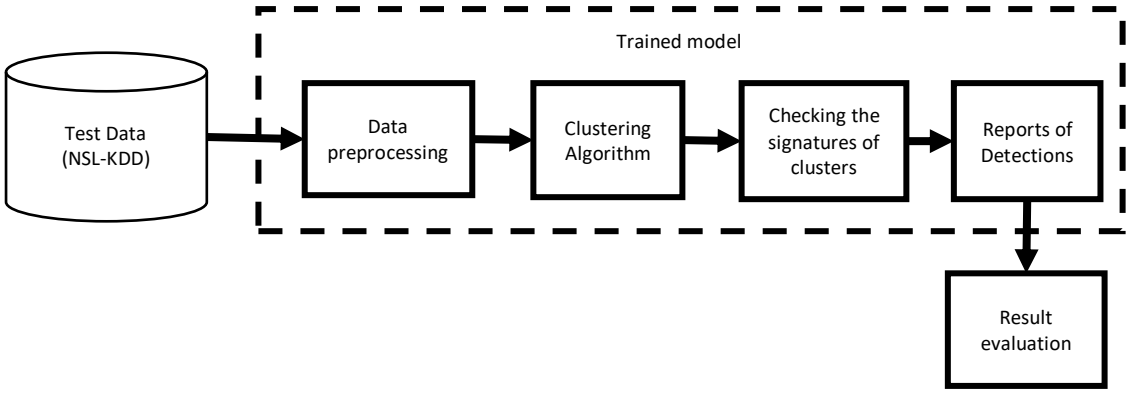

Figure 4. The model testing process.

Let k be the number of clusters formed after the clustering process, the clusters are $C_i,\ldots,C_k$, and the corresponding data set of the cluster is $D_i,\ldots,D_k$. Let V_i be the feature vector of the cluster C_i: $V_i=f(D_i)$

where f() is the function that computes the n most important features in the cluster, hence:
$V_i=[x_{i1},\ldots,x_{in}]$
where x_{in} is the nth most important feature in cluster C_i.

Algorithm 1 Training model

Input: train_data
A training model
Output: A set of feature vectors of anomaly clusters: S_a
1:D=train_data
2:S_a={}
3:(D_i,.., D_k)=clustering(D)
4:for (i=1; i<=k; i++) {
5: $V_i=f(D_i)$
6: if is_anomaly(C_i) then $S_a=S_a + V_i$
7: }
8: return S_a

Algorithm2 Anomaly detection

Input: test_data
The trained model with S_a
Output: number of anomaly clusters, detect
1:detect=0
2:S={}
3:(D_i,..., D_k)=clustering(D)
4:for (i=1; i<=k; i++) {
5:$V_i=f(D_i)$
6: if inside_Sa(V_i) then detect++
7: }
8: return detect

4.2.2. NSL-KDD Dataset

The NSL-KDD dataset [20] contains Internet traffic logs observed by a simple intrusion detection system and is likely to be encountered by an IDS. The dataset consists of 43 attributes in each record, with 41 related to the traffic itself, the last 2 being the label (attack or not) and the severity score of the input traffic. The training dataset consists of 125,973 records and the test dataset consists of 22,544 records. The training dataset contains 22 attack types and 17 more in the test dataset, classified into four groups DoS (Denial of Service), R2L (Remote to Local), U2R (User to Root), and Probe.

4.2.3. Data Preprocessing and Feature Extraction

In the 41 attributes of the input data set, there are 3 attributes of the categorical data type: protocol_type, service, and flag. The remaining attributes are numeric properties. The input data of the clustering algorithms in the model only includes numeric values, so to use the categorical attributes we transform the categorical attributes into numeric attributes by the One Hot Encoding technique.

The extracted data consists of many features and each feature has different units and magnitudes. This affects the efficiency of many algorithms, so it is necessary to adjust so that the features have the same data scaling. In the paper, MinMaxScaler and StandardScalar techniques are used to normalize data.

For the feature extraction method, Attribute Ratio [19] uses the OHE technique to convert attributes of categorical data types to numeric data. The Risk-based Acquisition feature extraction method [18] only uses the OHE technique to convert the protocol_type attribute to numeric data. The normalized and feature-extracted data set is divided into two parts: 80% of the dataset is used to train the model and the remaining 20% is a test dataset for model evaluation.

4.2.4. Model Training

We train the model in four cases using different feature extraction methods and clustering algorithms: RA with K-means (RA_K-means), RA with DBSCAN (RA_DBSCAN), AR with K-means (AR_K-means), and AR with DBSCAN (AR_DBSCAN). In each case, the training process analyzes and calculates the set of five important attributes with the highest rank based on the mean value in each data cluster. The training results identify sets of five attributes that are feature vectors of attack clusters used to detect attacks when applying the model. The set of five attributes with the highest rank in each cluster is the cluster feature vector mentioned above.

In the case of RA_K-means:

Applying the Elbow technique to select the optimal number of clusters k for the K-means algorithm, k=8 is determined. There are 8 clusters formed after the clustering process and the importance of the attributes in each cluster is shown in Figure 5. Table 1 lists the five most important attributes in each cluster.

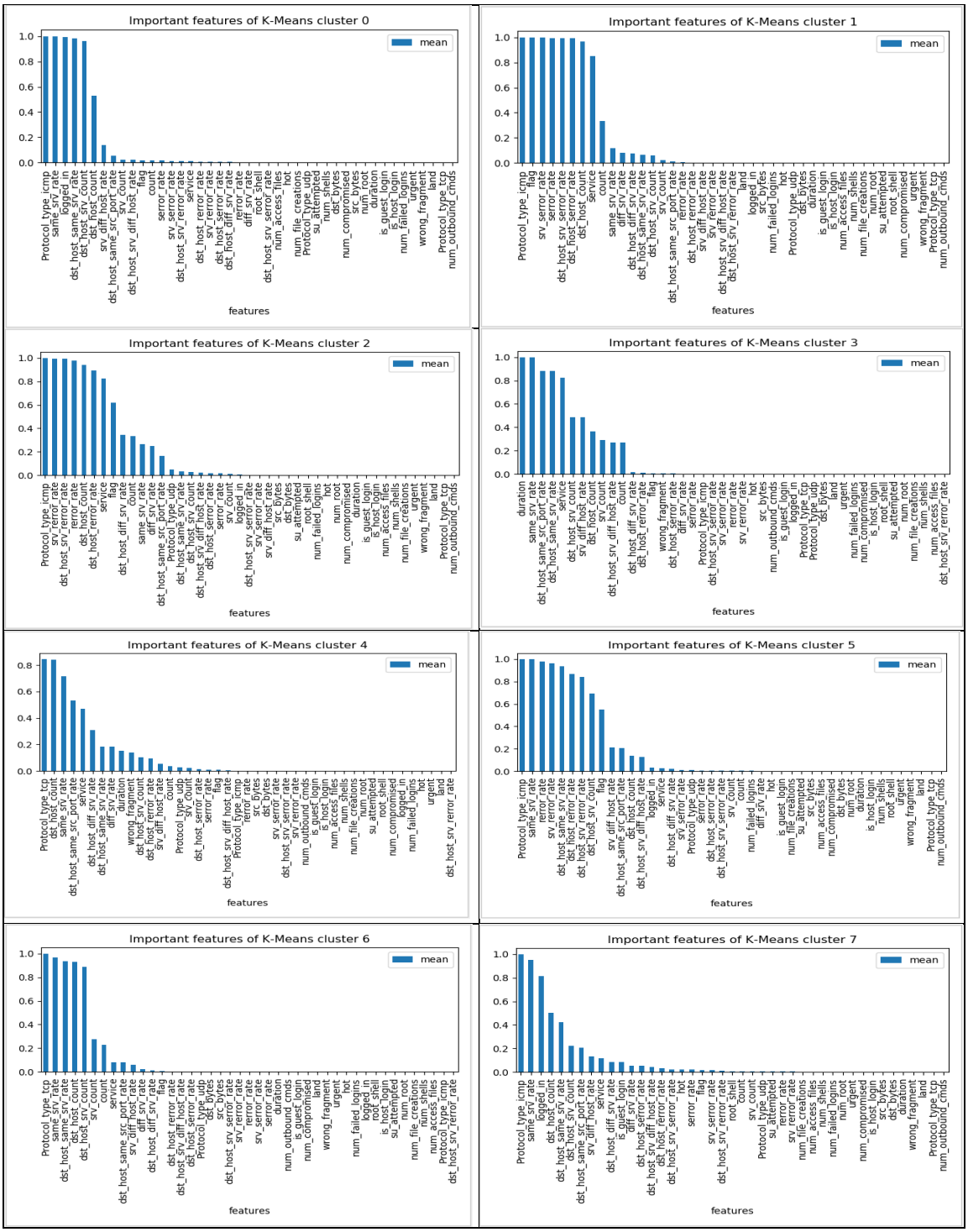

Figure 5. Graphs of the importance of attributes in clusters in the case of RA_K-means.

Table 1. Set of the five most important attributes of each cluster in case of RA_K-means.

Cluster0		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	same_srv_rate	Numeric
3	logged_in	Binary
4	dst_host_same_srv_rate	Numeric
5	dst_host_srv_count	Numeric

Cluster1		
Rank	**Feature**	**type**
1	Protocol_type_icmp	Nominal
2	flag	Nominal
3	srv_serror_rate	Numeric
4	serror_rate	Numeric
5	dst_host_srv_serror_rate	Numeric

Cluster2		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	srv_rerror_rate	Numeric
3	dst_host_srv_rerror_rate	Numeric
4	rerror_rate	Numeric
5	dst_host_count	Numeric

Cluster3		
Rank	**Feature**	**Type**
1	duration	Numeric
2	same_srv_rate	Numeric
3	dst_host_same_src_port_rat	Numeric
4	dst_host_same_srv_rate	Numeric
5	service	Nominal

Cluster4		
Rank	**Feature**	**Type**
1	Protocol_type_tcp	Nominal
2	dst_host_count	Numeric
3	same_srv_rate	Numeric
4	dst_host_same_src_port_rate	Numeric
5	service	Nominal

Cluster5		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	same_srv_rate	Numeric
3	rerror_rate	Numeric
4	srv_rerror_rate	Numeric
5	dst_host_same_srv_rate	Numeric

Cluster6		
Rank	**Feature**	**Type**
1	Protocol_type_tcp	Nominal
2	same_srv_rate	Numeric
3	dst_host_same_srv_rate	Numeric
4	dst_host_count	Numeric
5	dst_host_srv_count	Numeric

Cluster7		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	same_srv_rate	Numeric
3	logged_in	Binary
4	dst_host_count	Numeric
5	dst_host_same_srv_rate	Numeric

The crosstab method is used to calculate the probability of occurrence of normal data points or attack data points in clusters. The experiments use the pandas library to analyze the data of the samples in the cluster. The normal data type is labeled "0" and the attack data type is labeled "1", as shown in Figure 6.

clutering labels	0	1	2	3	4	5	6	7
0	2189	9	6221	459	20	1586	661	67
1	239	5826	118	19	2002	2	432	1145

Figure 6. Crosstab values in case of RA_K-Means

Based on the results of the crosstab analysis, we label each cluster by determining the distribution of data points and choosing the data type with the most frequency. Thereby identifying clusters 0.2,3,5 and 6 are normal clusters, and clusters 1, 4, and 7 are attack clusters. The corresponding sets of five attributes of the clusters are presented in Table 1. These attribute sets are the cluster feature vectors that the model relies on to detect attacks if they occur.

In the case of RA_DBSCAN:

The experimental parameters selected in the model applying the DBSCAN algorithm are eps=0.8, min_samples=850. There are 10 clusters formed after the clustering process and the importance of the attributes in each cluster is shown in Figure 7. Table 2 lists the five most important attributes in each cluster.

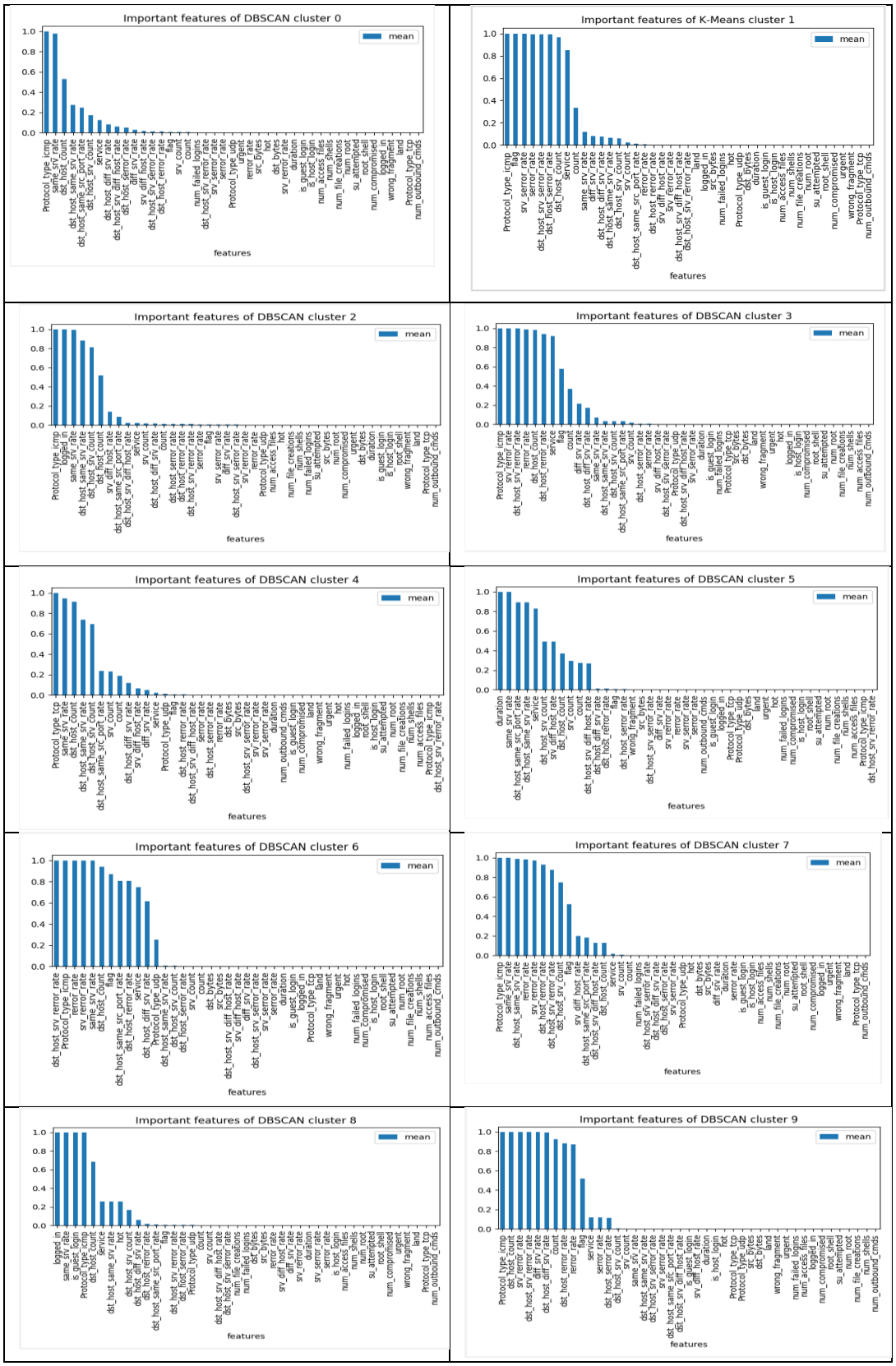

Figure 7. Graphs of the importance of attributes in clusters in the case of RA_DBSCAN.

Table 2. Set of the five most important attributes of each cluster in the case of RA_DBSCAN.

Cluster0		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	same_srv_rate	Numeric
3	dst_host_count	Numeric
4	dst_host_same_srv_rate	Numeric
5	dst_host_same_src_port_rate	Numeric

Cluster1		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	serror_rate	Numeric
3	srv_serror_rate	Numeric
4	dst_host_serror_rate	Numeric
5	flag	Nominal

Cluster2		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	logged_in	Binary
3	same_srv_rate	Numeric
4	dst_host_same_srv_rate	Numeric
5	dst_host_srv_count	Nominal

Cluster3		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	srv_rerror_rate	Binary
3	dst_host_srv_rerror_rate	Numeric
4	rerror_rate	Numeric
5	dst_host_count	Nominal

Cluster4		
Rank	**Feature**	**Type**
1	Protocol_type_tcp	Nominal
2	same_srv_rate	Numeric
3	dst_host_count	Numeric
4	dst_host_same_srv_rate	Numeric
5	dst_host_srv_count	Numeric

Cluster5		
Rank	**Feature**	***Type***
1	duration	Numeric
2	same_srv_rate	Numeric
3	dst_host_same_src_port_rate	Numeric
4	dst_host_same_srv_rate	Numeric
5	service	Nominal

Cluster6		
Rank	**Feature**	**Type**
1	dst_host_srv_rerror_rate	Numeric
2	Protocol_type_icmp	Nominal
3	rerror_rate	Numeric
4	srv_rerror_rate	Numeric
5	same_srv_rate	Numeric

Cluster7		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	same_srv_rate	Numeric
3	dst_host_same_srv_rate	Numeric
4	rerror_rate	Numeric
5	srv_rerror_rate	Numeric

Cluster8		
Rank	**Feature**	**Type**
1	logged_in	Nominal
2	same_srv_rate	Numeric
3	is_guest_login	Numeric
4	Protocol_type_icmp	Numeric
5	dst_host_count	Numeric

Cluster9		
Rank	**Feature**	**Type**
1	Protocol_type_icmp	Nominal
2	dst_host_count	Numeric
3	srv_rerror_rate	Numeric
4	dst_host_srv_rerror_rate	Numeric
5	diff_srv_rate	Numeric

Each cluster is also labeled as normal or attacked using the crosstab method as in the case of AR with K means above. Thereby identifying clusters 0,2,4,7,8 are normal clusters and clusters 1, 3, 5, and 6 are attack clusters. The corresponding sets of five attributes of these clusters are presented in Table 2. The corresponding attribute sets of the attack cluster are the attack cluster feature vectors, which are the signatures against which the model detects attacks.

The training process is similar for the other two cases AR_K-means, and AR_DBSCAN. In which we apply the Attribute Ratio feature extraction method. The results of the training also identify sets of five attributes that are indicative of attack activities so that the model can detect attacks if it does.

5. EXPERIMENTS

To evaluate the proposed attack detection method, we test four model cases in turn: RA_K-means, RA_DBSCAN, AR_K-means, and AR_DBSCAN. In each case, the test dataset is fed into the model for clustering. The feature vector of each cluster is determined and compared with the

known attack feature vectors in the trained model. If there is a similarity between the feature vector of the cluster under test and an attack feature vector, it is an indicator of an attack to warn. We proceed to label the predictions for the data points on the clusters, and the prediction label of each data point coincides with the predicted cluster label. Thereby the following parameters of the confusion matrix are calculated: TP (True Positive), TN (True Negative), FP (False Negative), and FN (False Negative). The model is evaluated based on the metrics: ACC (Accuracy), Sensitive or TPR (True Positive Rate), Precision or PPV (Positive Predictive Value), FPR (False Positive Rate), and F1 score. These metrics are calculated according to the formulas (5), (6), (7), (8), and (9) respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \quad (8)$$

$$F1\ Score = \frac{2TP}{2TP + FP + FN} \quad (9)$$

We implement the program and run it on a computer configured with Intel® Core™ i-3740 CPU@ 3.20 GHz, RAM: 16 GB. System type: 64-bit operating system, x64-based processor. Operating system: Windows 10 Pro 64-bit. IDE: Pycharm-JetBrains 2019.2.4 (Professional Edition) with Python 3. The program uses several libraries such as scikit-learn, scipy, numpy, joblib, pandas, and pyspark. The pandas is used to compute the mean values of the attributes in cluster 0 in the case of RA_K-means, as shown in Figure 8.

	count	mean	...	75%	max
Protocol_type_icmp	30762.0	1.000000e+00	...	1.000000e+00	1.000000
same_srv_rate	30762.0	9.988986e-01	...	1.000000e+00	1.000000
logged_in	30762.0	9.924907e-01	...	1.000000e+00	1.000000
dst_host_same_srv_rate	30762.0	9.808621e-01	...	1.000000e+00	1.000000
dst_host_srv_count	30762.0	9.586298e-01	...	1.000000e+00	1.000000
dst_host_count	30762.0	5.297252e-01	...	1.000000e+00	1.000000
srv_diff_host_rate	30762.0	1.373441e-01	...	1.700000e-01	1.000000
dst_host_same_src_port_rate	30762.0	5.410214e-02	...	3.000000e-02	1.000000
srv_count	30762.0	2.335092e-02	...	3.326810e-02	0.215264
dst_host_srv_diff_host_rate	30762.0	2.037026e-02	...	3.000000e-02	0.280000

Figure 8. The mean value of the attributes in the cluster 0 in the RA_K-means model.

The experimental process on the test dataset with the use of different feature extraction methods and different clustering algorithms has the following results:

-In the case of the RA withK-means model, three clusters are detected as anomalies.
-In the case of the RA with DBSCAN model detected four clusters as anomalies.
-In the case of AR with K-means model, three clusters are detected as anomalies

-In the case of RA with DBSCAN model detected four clusters as anomalies.
Setting up the confusion matrix for each case and calculating the evaluation metrics. The metrics including (ACC), Positive Predictive Value (PPV), true positive rate (TPR), false positive rate (FPR), and F1 score are presented in Table 3.

Table 3. Test results of four cases

Cluster algorithm	Feature extraction method	ACC (%)	PPV (%)	TPR (%)	FPR (%)	F1 (%)
K-means	Risk-based Acquisition	0.9742	0.9611	0.9917	0.0456	0.9762
DBSCAN	Risk-based Acquisition	0.9304	0.9830	0.8852	0.0176	0.9315
K-means	Attribute Ratio	0.9467	0.9999	0.9001	0.0001	0.9474
DBSCAN	Attribute Ratio	0.9626	0.9998	0.9296	0.0003	0.9634

The results in Table 3 show that when using RA, the results are good with K-means but not with DBSCAN. Whereas using AR is the opposite, the results are positive with DBSCAN but not good with K-means. In case AR-DBSCAN has accuracy over 96% and precision over 99%, F1 also achieves over 96%. In the four tested cases, the model using RA with K-means gives the best results, with an accuracy of more than 97% and other measures are all at a positive level, such as a high true positive rate, F1 Score also over 97%.

6. CONCLUSIONS AND FUTURE WORK

A lightweight method for detecting network attacks in large-scale traffic systems has been presented. The method exploits the utility of clustering techniques and detects attacks based on the feature vectors of the attack clusters. Thereby, the full manifestation of the attacks is determined and the actual attacks taking place in complex steps are controlled. Clustering quality is enhanced by data pre-processing and reasonable feature extraction methods. By using the proposed method, it is possible to implement a parallel-enabled detection system suitable for large-sized and high-traffic networks. How to implement such a parallel processing detection system has also been shown. Experiments give positive results, especially when using the model with the RA feature extraction method and the K-means clustering algorithm, the evaluation metrics all reach state-of-the-art. In the future, we will continue evaluating the method on other datasets, using new preprocessing methods along with other advanced clustering algorithms.

CONFLICT OF INTERESTS

The authors declare no conflict of interest.

REFERENCES

[1] Guan Xin and Li Yun-jie,(2010) “A new Intrusion PreventionAttack System Model based on Immune Principle”,International Conference on e-Business and InformationSystem Security (EBISS), in IEEE, pp. 1-4.

[2] A. H. Almutairi and N. T. Abdelmajeed, (2017) "Innovative signature based intrusion detection system: Parallel processing and minimized database", International Conference on the Frontiers and Advances in Data Science (FADS), pp. 114-119,DOI: 10.1109/FADS.2017.8253208.

[3] Khraisat A, Gondal I, Vamplew P, (2018) “An anomaly intrusion detection system using C5 decision tree classifier”, Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp. 149–155.

[4] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, (2013)"A survey of intrusion detection techniques in cloud", J Netw Comput Appl, vol. 36, no. 1, pp. 42–57.

[5] Wang. K and Stolfo.S.J, (2004) “Anomalous Payload-BasedNetwork Intrusion Detection”, 7th Symposium on RecentAdvances in Intrusion Detection, Volume 3224 ofLNCS., Springer-Verlag 203–222.

[6] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, (2012)"Using feature selection for intrusion detection system", International Symposium on Communications and Information Technologies (ISCIT), pp. 296–301.

[7] V. Jyothsna, and K. M. Prasad, (2019) "Anomaly-Based Intrusion Detection System", in Computer and Network Security. London, United Kingdom: IntechOpen, [Online]. Available: https://www.intechopen.com/chapters/67618 DOI: 10.5772/intechopen.82287

[8] Naqash, T., Shah, S.H. & Islam, M.N.U., (2022)” Statistical Analysis Based Intrusion Detection System for Ultra-High-Speed Software Defined Network”, Int J Parallel Prog 50, pp.89–114. https://doi.org/10.1007/s10766-021-00715-0

[9] Jisa David, Ciza Thomas,(2019) “Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic”,Computers & Security,Volume 82, pp. 284-295,ISSN01674048,https://doi.org/10.1016/j.cose.2019.01.002. https://www.sciencedirect.com/science/article/pii/S0167404818307624

[10] Sathish Alampalayam. Kumar et al., (2007)"Statistical based intrusion detection framework using six sigma technique," International Journal of Computer Science and Network Security, vol. 7, no. 10, pp. 35-44.

[11] N. A. Carreón, A. Gilbreath and R. Lysecky, (2020)"Statistical Time-based Intrusion Detection in Embedded Systems", Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 562-567, DOI:10.23919/DATE48585.2020.9116369

[12] Taher, K. A., Jisan, B. M., and Rahman, M. M., (2019) “Network intrusion detection using supervised machine learning technique with feature selection”, IEEE International Conference on Robotics, Electrical and Signal Processing Techniques, DOI:10.1109/ICREST.2019.8644161

[13] F. Hossain, M. Akter and M. N. Uddin,(2021) "Cyber Attack Detection Model (CADM) Based on Machine Learning Approach", 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 567-572, doi:10.1109/ICREST51555.2021.9331094.

[14] Ilhan Firat Kilincer, Fatih Ertam, Abdulkadir Sengur, (2021)“Machine learning methods for cyber security intrusion detection: Datasets and comparative study”, Computer Networks, Volume 188, ISSN 13891286, https://doi.org/10.1016/j.comnet.2021.107840. https://www.sciencedirect.com/science/article/pii/S1389128621000141

[15] Khushnaseeb Roshan and Aasim Zafar, (2021) “Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with Shapley Additive Explanation (SHAP)”, International Journal of Computer Networks & Communications (IJCNC) Vol.13, No.6, November 2021, pp.109-128, ISSN:0974-9322 (Online); 0975-2293(Print), https://doi.org/10.5121/ijcnc.2021.13607

[16] Venkata Ramani Varanasi et al., (2020) “A Comparative Evaluation of supervised and unsupervised algorithms for Intrusion Detection”, International Journal of Advanced Trends in Computer Science and Engineering, 9(4), pp. 4834 – 4843.

[17] Karbal Basma and Romadi Raha, (2020) “A Comparison of Different Machine Learning Algorithms for Intrusion Detection”, International Conference on Advanced Communication Systems and Information Security-ACOSIS, November 2020

[18] J Juanchaiyaphum, N Arch-Int, S Arch-Int, S Saiyod, (2014) “Symbolic Data Conversion Method Using The Knowledge-Based Extraction In Anomaly Intrusion Detection System”, Journal of Theoretical & Applied Information Technology, Vol. 65 No.3, ISSN:1992-8645, E-ISSN: 1817-3195, pp. 695-701.

[19] Hee-su Chae, Byung-oh Jo, Sang-Hyun Choi, Twae-kyung Park, (2013) “Feature Selection For Intrusion Detection using NSL-KDD”, Recent Advances in Computer Science, pp184-187.

[20] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, Ali A. Ghorbani, (2009) “A Detailed Analysis of the KDD CUP 99 Data Set”, Proceedings of the Second IEEE International Conference, DOI: 10.1109/ CISDA.2009.5356528, pp.53-58

[21] Binita Bohara et al, (2020) “Survey On The Use of Data Clustering for Intrusion Detection System in Cybersecurity”, International Journal of Network Security & Its Applications (IJNSA) Vol. 12, No.1, January 2020, 12(1): 1–18. DOI:10.5121/ijnsa.2020.12101.

[22] Md Tahmid Rahman Laskar, et al.,(2021) “Extending Isolation Forest for Anomaly Detection in Big Data via K-Means”, ACM Trans. Cyber-Phys. Syst. 5, 4, Article 41 (October 2021), 26 pages. https://doi.org/10.1145/3460976

[23] Fargana J. Abdullayeva,(2022) “Distributed denial of service attack detection in E-government cloud via data clustering”, Array, Volume 15,2022,100229, ISSN 2590-0056, https://doi.org/10.1016/j.array.2022.100229. https://www.sciencedirect.com/science/article/pii/S2590005622000686

[24] Markus Netzer, Jonas Michelberger, Jürgen Fleischer,(2020) “Intelligent Anomaly Detection of Machine Tools based on Mean Shift Clustering”,Procedia CIRP,Volume 93,2020,ISSN 2212-8271, pp. 1448-1453, https://doi.org/10.1016/j.procir.2020.03.043. https://www.sciencedirect.com/science/article/pii/S2212827120306454

[25] M. J. Middlemiss and G. Dick, (2003) "Weighted feature extraction using a genetic algorithm for intrusion detection", The Congress on Evolutionary Computation(CEC '03.), Vol.3, pp. 1669-1675, DOI: 10.1109/CEC.2003.1299873.

[26] Sergii Lysenko, O. Savenko, K. Bobrovnikova, and A. Kryshchuk, (2018) “Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks”, Communications in Computer and Information Science, pp. 385-401.

[27] Sergii Lysenko et al., (2020) “Detection of the botnets’ low-rate DDoS attacks based on self-similarity”, International Journal of Electrical and Computer Engineering (IJECE), Vol 10, No 4 August2020,p-ISSN 2088-8708, e-ISSN 2722-2578, pp. 3651-3659, http://doi.org/10.11591/ijece.v10i4.pp3651-3659.

[28] Lloyd, S. P., (1957) Least squares quantization in PCM. Technical Report RR-5497, Bell Lab, September 1957.

[29] Ester, Martin; Kriegel, Hans-Peter; et al.,(1996) “A density-based algorithm for discovering clusters in large spatial databases with noise”, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.

AUTHORS

Nguyen Hong Son received his B.Sc. in Computer Engineering from The University of Technology in HCM city, and his M.Sc. and Ph.D. in Communication Engineering from the Post and Telecommunication Institute of Technology Hanoi. His current research interests include communication engineering, machine learning, data science, network security, and cloud computing.

Ha Thanh Dung, received his B.Sc in Information Technology from VNU Hanoi-University of Science, and his M.Sc in Data Transmission and Computer Networks from Post and Telecommunication Institute of Technology in 2012. His research areas are communication engineering, information systems, machine learning, and network security