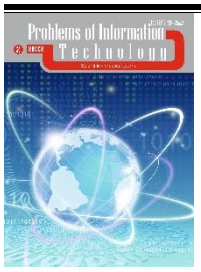

Available online at www.jpit.az

13 (2)
2022

INVESTIGATION OF CLUSTERING AND CLASSIFICATION METHODS FOR INTELLECTUAL ANALYSIS OF LOG FILES

Babak R. Nabiyev [a], Fuad I. Ahmadov [b]

[a,b] Institute of Information Technology, Azerbaijan National Academy of Sciences, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

[a] babek@iit.science.az; [b] f.ahmedov@iit.science.az

ARTICLE INFO

http://doi.org/10.25045/jpit.v13.i2.05

Article history:
Received 25 January 2022
Received in revised form 15 March 2022
Accepted 19 May 2022

Keywords:
Log file
K-means
CURE algorithm
Naive Bayes
EM algorithm
kNN
Decision tree
KDD CUP 99

ABSTRACT

Today, the application of information technology in all areas of our lives has led to wider spread and popularity of cybercrime. In modern industrial control systems and cyber-physical systems, log files are very important in terms of detecting cyber incidents, identifying and preventing threats and anomalies. However, today, a large volume of log files generated in these systems greatly complicates the process of extracting useful information from them. This, in turn, highlights the need for intellectual analysis of log files. To this end, this article explores a number of clustering and classification methods and algorithms for the intellectual analysis of log files. Thus, K-means, CURE, EM, kNN, Naive Bayes and DT algorithms are selected out of these algorithms and their working principle is studied, explained, and the application of each algorithm on KDD CUP 99 data set is studied and compared.

1. Introduction

Currently, information technologies have entered all spheres of our lives, and especially in recent times, in the face of the new realities created by the covid-19 pandemic, application of the digital version of all traditional solutions has become forced. In addition, technological innovations brought by the 4th industrial revolution can be cited as an example. In connection to abovementioned, we are witnessing even wider spread and popularization of malicious acts in the digital environment. Proceeding from these, we can confidently say that as a result of emerging threats, and in order to legally find the source of these threats and bring the perpetrators to justice, there is a great need for intellectual analysis of log files as a method of detecting cybercrimes.

As in other computer systems, industrial control systems (ICS) generate various type and large volumes of log files. These log files are the data allowing to monitor the status and security of the ICS that generate them, and containing useful information. In other words, log files are very important in terms of detecting and eliminating problems on IMS (Zwietasch, 2014). As we know, log files are a type of big data (BD). This means that analysis of log files is quite a complex process (Zulfadhilah, Prayudi, Riadi, 2016). In terms of detecting and eliminating mentioned cyber incidents, extraction and analysis of valuable information from this data is immensely needed. Based on abovementioned, in the article, intellectual data analysis methods and algorithms for analysis of log files are researched and comparative analysis of each of them is conducted.

Intellectual data analysis is a field of science on extraction process of hidden and useful data from unprocessed BD, and it is currently applied in different spheres of human life. Many methods used in intellectual data analysis combines methods used in machine learning, statistics and artificial intelligence fields (Aliguliyev, Niftaliyeva, 2016). This study explores machine learning methods for intellectual analysis of log files.

Machine learning is a field of artificial intelligence implementing data analysis using special algorithms on bases of BD (Shikhaliyev, 2022). In other words, instead of manually integrating a program using a set of different commands to perform a set of tasks on a given machine, it is a method of teaching one or more machines to perform these tasks through a large amount of data and algorithms.

Usually machine learning methods are grouoped into 4 categories depending on research object or problem statement (Pecht, Myeongsu, 2018):

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning
4. Reinforcement learning

Classification and clustering methods used in supervised learning and unsupervised learning methods for intellectual data analysis are studied.

Application of clustering and classification algorithms in intellectual analysis of BD is considered a convenient method in terms of identifying clusters and classes and extracting valuable information from those data. Clustering and classification algorithms such as *K*-means, CURE (Clustering Using Representatives), *Expectation* and Maximization (EM), k-nearest neighbors (kNN), NB (Naive Bayes) and decision tree (DT) were used for log analysis. Advantages and disadvantages of these algorithms are compared, taking into account factors such as performance, accuracy, time and memory resource usage, etc.

2. Clustering algorithms and their operating principle

Clustering is an unsupervised learning method of machine learning. Here, data that are as similar as possible to each other are collected in the same cluster, and dissimilar data are collected in different clusters. One of the main features of this method is that the clusters to be identified are not known in advance. The main goal of clustering is to minimize the distance between data in the same cluster and maximize the distance between clusters. As a result, the process of extracting useful information from large volumes of data becomes quite easy (Kamber & Pei, 2011). The mentioned analysis process is performed on the basis of k-means, EM and CURE algorithms.

2.1. K-means algorithm

K-means algorithms divides n number of data by k number of cluster in d-dimensional Euclidean space and $X = \{x_1 \dots , x_n\}$ data set (Nabiyev, 2015). As noted above, the goal here is to ensure that within-cluster similarities are at a maximum level and inter-cluster similarities are at a minimum level in the clusters obtained after the clustering operation is implemented.

2.1.1. Operating principle

K-means is a simple in terms on operating principle, and widely used algorithm. Thus, this algorithm allows to rapidly and effectively cluster large volumes of data. Here "*k*" is a quantity representing the number of clusters implemented and identified before the algorithm starts. Assigning a value to *k* is one of the main disadvantages of this algorithm. That is, an approach called "Elbow method" is applied in order to eliminate this disadvantage and determine the k-number (Umargono, Suseno, Gunawan, 2019) (Fig. 1).

The curves depicted in Figure 1 are the distance between data within the same cluster and at maximum distance from each other. In other words, it is the similarity index of within-cluster objects (Figure 1). As the figure illustrates, the value of k is directly proportional to the quality of clustering. Particularly, the number of clusters increase with the increase in the value of k-number, which in its turn, increases the quality of clustering. However, in addition to the quality, the effectiveness of the algorithm in terms of speed, time, memory resources, etc. should also be taken into account. Therefore, the process ends at the point where no sharp increase is observed and k is assigned that value (Fig. 1).

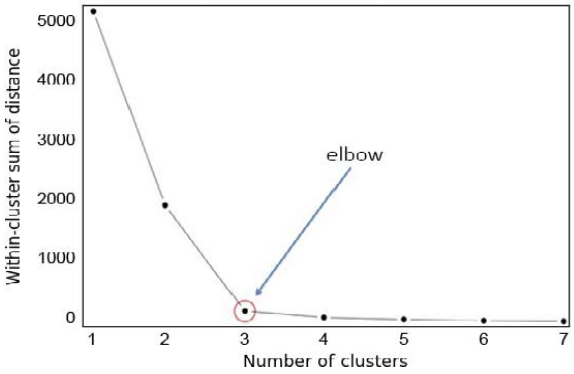

Fig. 1. A graphic description of the Elbow method

At the initial stage, the clustering process is implemented by random placement of k-number of cluster centers in the data space. Then the distance between each data and those cluster centers is calculated. In this case, different distance measurements can be used. These include Euclidean, Manhattan, Minkovski, Cosine similarity etc. Here, in d-dimensional space, in order to calculate the distance between $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ points, Euclidean metric is used and its formula is provided below (Nabiyev 2015):

$$dist(X,Y)\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

Here, the concept of distance is characterized by the degree of similarity property of the data.

In the next step, each data refers to its nearest cluster center or cluster. After the completion of this process, the mean value of the data within each cluster is found as the last step, and the cluster centers selected in the initial step change their positions and are located at that point.
In the space, a mean value of given points is calculated using following formula:

$$m = \frac{1}{n}\sum_{i=1}^{n}(x_i,y_i) \qquad (2)$$

Where m – represents the center of data, n –the quantity of data. In two-dimensional space, the coordinates of the points are marked with x_i and y_i parameters.

Since *K*-means is an iterative algorithm, the abovementioned last 2 steps are repeated until convergence, i.e. until the best clusters are found in terms of optimizing the result. This, in turn, ensures that the desired result is achieved in the end.

2.1.2. Advantages and disadvantages

K-means algorithm has several advantages and disadvantages. Main advantages of the algorithm are listed below:

- Rapid execution of operations, even in a set of large volume of data
- It is scalable, i.e. possible to use in data growing in volume
- Very good results in clear and prominent data sets
- Easy to apply
- Availability of algorithm code in many programming languages

Biggest disadvantage of k-means algorithm is that, it depends on selection of cluster centers randomly defined in the beginning. Thus, the quality of obtained result is directly dependent on selection of these cluster centers. In addition, the complexity of correctly determining the k number and the inability to cluster outliers are the main disadvantages of this algorithm (Alguliyev, Aliguliyev & Sukhostat, 2020).

2.2. CURE algorithm

The second method to be analyzed is the CURE algorithm. The main reason for the development and application of this algorithm was a number of disadvantages of previously used clustering algorithms. For example, they are unable to cluster non-spherical datasets and are vulnerable to outliers (Guha, Rastogi & Shim, 2001). Based on these, the CURE algorithm is proposed as a new approach to improve the quality of clustering.

2.2.1. Operating principle

Unlike k-means method, in CURE algorithm created clusters are represented with a constant number of data i.e., representatives without "cluster center" parameter. Algorithm continues until desired number of clusters is achieved by merging clusters with nearest representatives. In order to find newly created cluster representatives at each stage, the representatives of each of the merging clusters are multiplied by the numerical value of the parameter called shrink factor (α) and brought closer to the cluster center. Thus, the distance between the representatives of the cluster and the center of the cluster decreases α-fold. Hence, obtaining optimal clusters depends on three parameters: number of clusters (k), number

of representatives (c) and shrink factor (α). The sequence of steps performed during the execution of the algorithm is as follows (Demiralay & Çamurcu, 2005):

1. In the first step, k number is defined as the input parameter.
2. For each cluster, a fixed number and c number of representatives located at a maximum distance from each other within cluster are selected.
3. Distance between clusters is determined by calculating the Euclidean distance between separate cluster representatives.
4. Nearest cluster pairs are merged.
5. In order to find the representatives of the newly formed clusters, we select c number of representatives of sub-clusters composing this cluster that are closest to the center. Then, these points are brought closer to the center by α – fold.
6. Steps 3, 4 and 5 are repeated until the number of clusters reaches the k number entered as input parameter.

In order to find optimal clusters in the CURE algorithm, experiments conducted on different data sets were analyzed in MATLAB software by assigning different values to three main parameters such as cluster number (k), representative number (c) and shrink factor (α) (Demiralay & Çamurcu, 2005).

Analysis of effect of cluster quantity (k) on clustering in the algorithm: In CURE algorithm k parameter is considered as the completion condition of clustering process. In order to clearly see the effect of k parameter on clustering, CURE algorithm is accepted as k=3, 4, 5 for circular data sets and k=4, 6, 8 for square data sets. Here, in order to clearly observe the effect of k on the algorithm, c and α parameters are kept constant. Cluster representatives are given as small squares within provided sets. (Demiralay & Çamurcu, 2005) (Fig. 2).

In Figure 2, the results of this process are shown for different values of the k number, accepting as α=0.2 and c=10 on the circular data set of the CURE algorithm and keeping their values constant. In data clustering process for k=3 value, 2nd and 4th data sets are merged, ideal clustering is obtained when k=4, and the biggest data set is divided, creating a new cluster when k=5. In the square data set, for different values of the k parameter α=0.3 and c=10 are accepted (Fig. 2). When k=4, then 1st, 2nd, 3rd, 4th and 6th square groups are merged and formed the 1st cluster, 5th, 7th and 8th groups are correctly clustered and relevantly formed 2nd, 3rd and 4th clusters. In accordance with k=6 value, 1st and 3rd groups create the first cluster, 4th and 6th groups form the third cluster. Other clusters are correctly found. Optimal results for this data set is possible at k=8 value (Demiralay & Çamurcu, 2005).

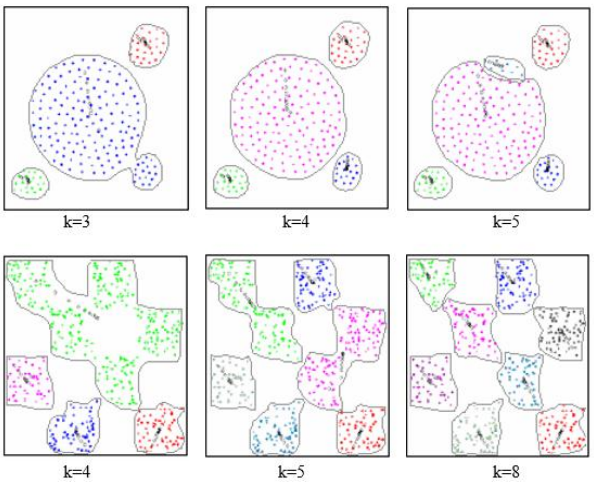

Fig. 2. Clusters formed when α=0.2, c=10 and k=3, 4, 5 are selected in the circular dataset and α=0.3, c=10 and k=4, 6, 8 are determined in the square dataset in CURE algorithm

Analysis of the effect of the number of representatives (c) on clustering in the algorithm:

In CURE algorithm, number of representatives is the quantity of data used to represent one cluster. Here, α and k parameters are kept constant, and clustering results are obtained for different values of c. In Figure 3, for circular data set α=0.2 and k=4 is accepted and clustering results for c=4, 10, 20 values is demonstrated. Here, ideal clustering is observed at c=10 value (Figure 3). Overall, when c parameter has a very small value, CURE algorithm operates as a cluster center principled algorithm and successful results cannot be obtained. Because, non-spherical clusters cannot be found in this instance. On the other hand, when c number has a very high value, algorithm tries to find enlarged i.e., large sized clusters. This, in turn, results in slower operation of algorithm in terms of speed and merging of separated datasets with shared neighboring points. In this case, the quality indicator of clustering is reduced (Demiralay & Çamurcu, 2005).

Fig. 5 shows the results obtained for values of c=2, 5, 10 when α=0.3 and k=8 are given in the square data set. Here it is clearly seen that the clusters are ideally organized when c=10.

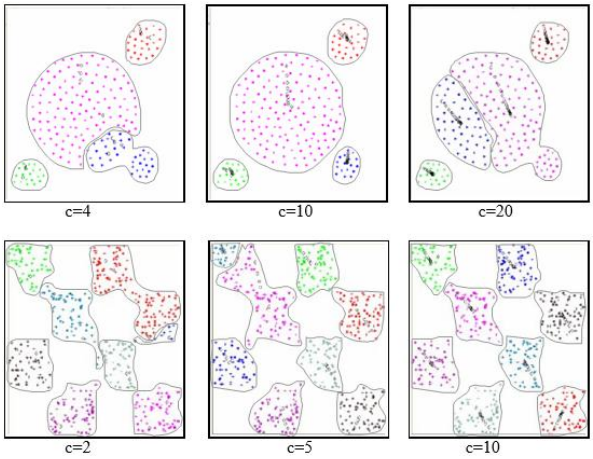

Figure 3. Clusters formed when α=0.2, k=4 and c=4, 10, 20 are chosen in circular dataset and α=0.3, k=8 and c=2, 5, 10 accepted in square dataset in CURE algorithm.

Analysis of the effect of shrink factor parameter (α) on clustering in the algorithm:

In order to analyze the effect of α parameter on clustering in this algorithm, the results of the experiments performed on circular and square datasets are depicted in Figure 4 and Figure 5, respectively. In Figure 4, k=4 and c=10 is accepted for circular clusters and results of clustering process for α=0.1, 0.2, 0.3 and 0.9 values; and in Figure 5, k=8 and c=10 is used for square clusters and results of clustering process for α=0.1, 0.2, 0.3 and 0.9 values are demonstrated. If attentively inspected, images show that for value of α=0.1, the cluster representatives are located far away from the centers of those clusters, so the desired result is not achieved. On the other hand, smaller the value of the α parameter, the more stable this algorithm is against outlier data. Consider these, we can see that optimal clustering is achieved at α=0.2 for circular data sets and α=0.3 values for square data sets (Demiralay & Çamurcu, 2005).

As seen from the examples, it is quite important to correctly select a representative, number of clusters or shrink factor number in order to successfully find the clusters in CURE algorithm.

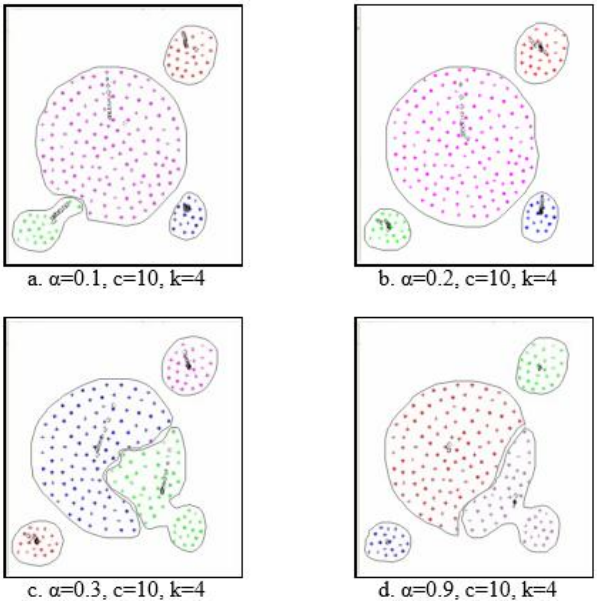

Figure 4. Results of application of CURE algorithm performed with different α values in a circular data set.

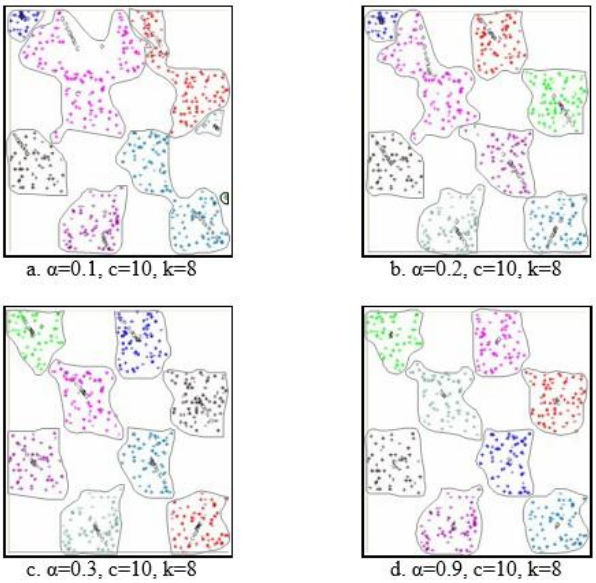

Figure 5. Results of application of CURE algorithm performed with different values of α in a square data set.

As seen from the examples, it is quite important to correctly select a representative, number of clusters or shrink factor number in order to successfully find the clusters in CURE algorithm.

2.2.2. Advantages and disadvantages

The main advantage of the CURE algorithm is that it can cluster arbitrary-shaped datasets, i.e., both spherical and non-spherical datasets with high accuracy. In addition, since the operating principle of the algorithm is different from others,

CURE is highly resistant to outliers. Another major advantage is that it allows for good scaling for large databases (Guha, Rastogi & Shim, 2001; Demiralay & Çamurcu, 2005).

On the other hand, obtaining an ideal result in the algorithm depending on the correct selection of the value of parameters *k*, c and α, and the complexity of this process is considered to be the main disadvantage of this algorithm (Guha, Rastogi & Shim, 2001; Demiralay & Çamurcu, 2005).

2.3. EM algorithm

EM algorithm is the next method to be analyzed. EM algorithm is characterized by evaluation of parameters given for maximization of probability of observed data. That is, instead of using a distance criterion to determine in which cluster a data will be located, it uses prediction criteria.

EM algorithm is an effective method widely used in many studies, especially recently. EM is an iterative algorithm that performs maximum likelihood forecasts in order to solve missing data problems. Each iteration of the algorithm consists of two stages. These stages are called finding expectation (E-stage) and maximization (M-stage) (Ömürbek, Dağ & Eren, 2020). Particularly, first of all, the best likelyhood of obtaining hidden data are estimated by looking at the parameters of the observed data at E-stage. Results obtained at E-stage are entered as input parameters at M-stage, and likelyhood of hidden data entered at M-stage belonging to one or more clusters is calculated and this probability is maximized. Results obtained from M-stage are entered as an input parameter to E-stage, and the previously explained process is repeated until the error indicator of the algorithm drops below a certain percentage.

Overall, there are two approaches to completion of iteration process: (Zafer, 2006):

1. First approach is determination of number of iterations in advance. However, his approach may not always provide correct results. That is, optimal result for various data is not achieved with standart number of iterations.
2. The second approach is stating conditions for the problem in advance. That is, obtained result is compared to the value determined in the problem statement. If obtained result is smaller than that value, then iteration process is completed.

2.3.1. Operating principle

The EM algorithm is particularly applicable to data sets of incomplete or unknown type. Application of the maximum likelihood EM algorithm to many statistical models was implemented by Dempster, Laird, and Rubin in 1977 (Ömürbek, Dağ & Eren, 2020). Algorithm is explained as following:

Assume that Ωx is an incomplete data space, x is an incomplete data vector, $f(x \mid \theta)$ is the likelihood function for x and Ωy is completed data space, y is incomplete data vector, $g(y \mid \theta)$ is the likelihood function for y. Here, we assume that there is a $y \rightarrow y(x)$ form of connection from Ω_x to Ω_y and a likelihood function for this is expressed as follows (Zafer, 2006):

$$g(y \mid \theta) = \int_{\Omega_{y(y)}} f(x \mid \theta) \qquad (3)$$

where $\Omega_{y(y)}$ is the subspace of Ω_x expressed with $y \rightarrow y(x)$ equation.

$LL_C(\theta) = log\ f(x \mid \theta)$ function is a probability log function consisting of complete data and $LL_C(\theta) = log\ g(y \mid \theta)$ is a probability log function that consists of incomplete data. Likelihood log function stands for the logarithmic calculation of the likelihood of data belonging to clusters. Objective of EM algorithm is to find the maximum likelihood estimation (MLE) of θ. Here, θ is an unknown parameter vector searched to find the MLE. Maximum likelihood estimations are obtained by solving the $LL_C(\theta)$ likelihood log function as a result of iterations.

$$\theta^{(k+1)} = \arg\max_{\theta \epsilon \Theta} E\left[LL_C(\theta) \mid y, \theta^{(k)}\right] \qquad (4)$$

Formula (4) is divided into two parts - steps E and M. In step E, conditional mathematical expectation and k^{th} temporary values of the parameters are calculated based on the likelihood log function for the observed data:

$$\Phi(\theta \mid \theta^{(k)}) = E\left[LL_C(\theta) \mid y, \theta^{(k)}\right] \qquad (5)$$

In step M, value $\theta^{(k+1)}$ that maximizes the $\Phi(\theta \mid \theta^{(k)})$ calculated in step E is found:

$$\theta^{(k+1)} = \arg\max_{\theta} \Phi(\theta \mid \theta^{(k)} \quad (6)$$

Iteration continues until $LL(\theta^{(k+1)}) - LL(\theta^{(k)})$ difference, i.e. similarity difference between steps E and M reaches a minimum.

2.3.2. Advantages and disadvantages

The main advantage of the EM algorithm is its sustainability. Thus, it makes decisive estimations until the optimal result is obtained. After each iteration process, it is inevitable that the likelihood of recovery of hidden data will increase. In addition, it can be easily implemented despite the limitation in parameters (Zafer, 2006).

The main shortcoming of the algorithm is very slow convergence when there is too much missing data, that is, the likelihood maximization process requires a long time. In addition, reduced stability against the values of initial parameters compared to other algorithms is also considered one of the main disadvantages of the EM algorithm (Zafer, 2006).

3. Classification algorithms and their operating principles

In machine learning, classification method means classification of objects or data into pre-divided groups. Classification is the supervised learning method of machine learning. This method classifies data into determined groups based on one or several common characteristics, and provides analysis of this data (Kesavaraj & Sukumaran, 2013). Classification algorithms such as kNN, NB, DT are studied, analyzed and compared for intellectual analysis of log files.

3.1. kNN algorithm

kNN is one of the most widely and commonly used algorithms in classification method. The kNN algorithm, considered one of the simplest algorithms in machine learning, is used for data classification. In addition, it is also used for regression. This algorithm classifies based on the similarity between the selected object and the closest object or objects to that object.

3.1.1. Operating principle

Operating principle is quite simple. The classified object or data is assigned to one of the previously known classes. At this time, data selects and classifies k number of points closest to itself. (Ying et al., 2021). This process is explained more clearly below:

1. Determination of the value of *K* parameter. Here, *k* number of nearest point must be reviewed when we say nearest neighboring points of the given point. For example, if *k*=5 is determined, then nearest 5 points are reviewed.
2. Calculation of the distance to the nearest points. At this stage, the distance between classified data and its nearest data is calculated. The distance is calculated using different methods. The most commonly used methods are Euclidean, Manhattan and Minkowski distances.
3. After finding the *k*-number of nearest neighboring points of the given point using one of the abovementioned formula, the point is assigned to the group where number of class data prevails. Here, if there are 2 classes, a single number is assigned to k parameter in order to avoid equality.

$$dist(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^{p})^{\frac{1}{p}} \quad (7)$$

Mathematical expression of Minkowski distance for finding the distance between two points is demonstrated (*formula 6*). Here, if *p*=2, then we obtain Euclidean distance, if *p*=1 then we obtain Manhattan distance. Mathematical expression of Manhattan distance is as following:

$$dist(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad (8)$$

Operating principle of kNN algorithm is more clearly explained in Figure 6 (Altunkaynak, Başakın, Kartal, 2020).

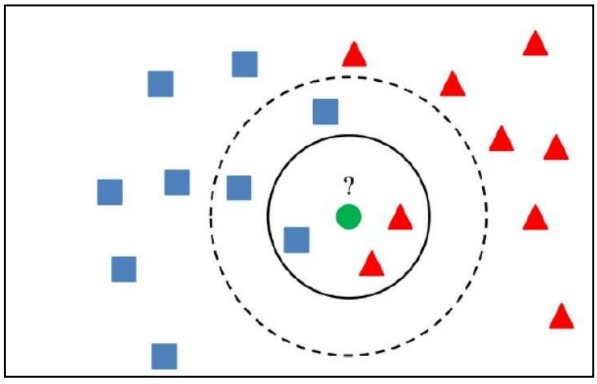

Figure 6. An example of kNN classifier.

Figure 6 depicts that a new object is included in the data set and this data is divided into 2 classes. The goal here is to determine the optimal number of neighbors (*k*). If *k*=3, this object belongs to the triangular data class, and if *k*=5, to the square data class. In many problems of this type, choosing the optimal value of *k* is quite complicated. Because, when the value of *k* parameter is changed, the class which the included object belongs to also changes. This leads to inaccurate results.

In general, if the value of k is very small, noisy data in the dataset will change the outcome of this problem significantly. On the other hand, giving a very high value to k also leads to slow operation of the algorithm and high use of computing and memory resources. This, in turn, lowers the quality indicator.

The solution of this problem is quite complex. That is, despite the proposal of several approaches, desired outcome is not achieved. The most optional proposed solution method is following (Nadkarni & Prakash 2016):

$$k = \sqrt{n} \tag{9}$$

Here, *k* is the number of neighboring elements and n is the number of trained elements.

3.1.2. Advantages and disadvantages

Main advantage of this algorithm is simplicity of operating principle and interpretation of the algorithm, the possibility to change the algorithm by using the most suitable combination functions and metrics adjusting the algorithm for a specific task (Wang & Li, 2010).

As for its shortcomings, the biggest disadvantage of the kNN algorithm is the difficulty to choose the correct number of *k*. Because the performance of the algorithm directly depends on the selection of the parameter *k*. Besides, the reduction of the operation speed of the algorithm in large volumes of data, use of high memory resources, and late completion of the process in terms of time are among main disadvantages of this algorithm. (Wang & Li, 2010).

3.2. Naive Bayes algorithm

Naive Bayes algorithm is an effective classification algorithm widely used in machine learning and intellectual data analysis. Operating principle of the algorithm is based on Bayes theorem (Abdullayeva & Ojagverdiyeva, 2021; Yang, 2016).

Bayes theorem was developed by Thomas Bayes who lived in 1701-1761. This theory provides the connection between prior and posterior probabilities in probability distribution for a random variable. Bayes theorem is expressed using following formula (Abdullayeva & Ojagverdiyeva, 2021; Yang, 2016)

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)} \tag{10}$$

- $P(A|B)$ – Probability of A event is occuring given B is implemented
- $P(B|A)$ - Probability of B event is occuring given A is implemented
- $P(A)$ and $P(B)$ – Prior probabilities of A and B events

In the Naive Bayes classifier, we know in advance, to which class the groups and training data belong to. An example of this is determining whether an arbitrary request sent to the network is an attack or not. For this example, the classes in the learning set can be in the form of "attack" and "normal" queries. Naive Bayes algorithm calculates the probability of all situations for given objects and classifies these objects according to the highest probability value. That is, using Bayes theorem the probability of all classes that classified data can belong to is determined, then the class with biggest posterior probability is selected (Abdullayeva & Ojagverdiyeva, 2021; Yang, 2016).

3.2.1. Operating principle

Let's assume that $X = \{x_1, x_2, \dots, x_n\}$ data set is provided and let's hypothesize that $C = \{C_1, C_2, \dots, C_m\}$ class set is known in advance. In this case, probability of object x_i belonging to class C_j is as following according to formula (9) (Imamverdiyev & Nabiyev, 2014):

$$P(C_j|x_i) = \frac{P(C_j)\,P(x_i|C_j)}{P(x_i)} \tag{11}$$

Where $P(C_j)$ – prior probability of data x_i belonging to class C_j and determined based on definition provided below:

Decision function for finding posterior probability in NB classifier is as following:

$$d(x_i) = \underset{c \in C}{arg\,max}\, P(c|x_i) \tag{12}$$

It is possible to convert the decision function shown in (10) based on Bayes theorem:

$$d(x_i) = \underset{c \in C}{arg\,max}\, P(c|x_i)$$

$$= \underset{c \in C}{arg\,max}\, \frac{P(x_i \mid c)\, P(c)}{P(x_i)} \quad (13)$$

$$= \underset{c \in C}{arg\,max}\, P(x_i \mid c)\, P(c), \text{ where } P(x_i) - \text{ is constant.}$$

As it is hypothesized that attributes are not interdependent in NB algorithm, calculation of $P(x_i|c)$ is simplified and can be written as following:

$$P(x_i|c) = \prod_{i=1}^{n} P(x_i \mid c) \quad (14)$$

At the end, decision function of NB classifier is as following:

$$d(x_i) = \underset{c \in C}{arg\,max} \prod_{i=1}^{n} P(x_i \mid c) \quad (15)$$

3.2.2. *Advantages and disadvantages*

NB algorithm is stable against incomplete data and is very simple in comparison with other classification algorithms. Since, it is sufficient to learn training database only once and highly accurate results are provided with a small number of data. In addition, the NB classifier is preferred for classifying large datasets, especially text-type data as it is linearly scalable (Abdullayeva & Ojagverdiyeva, 2021; Dilber, 2020).

The most important disadvantage is that "zero probability" can occur. Zero probability is the case when searched sample is not found the dataset. The simplest method of avoiding it consists of assigning a minimal value to all data. Usually, this method is called Laplas method. In addition, connections between variables cannot be modeled, as it is hypothesized that their characteristics are not interdependent (Dilber, 2020).

3.3. *Decision tree algorithm*

The decision tree algorithm is one of the algorithms used in the classification and regression methods of machine learning and considered effective for the intelligent analysis of data. Decision tree is a method used to divide large data into small classes by passing them through a series of decision-making (question-and-answer) stages. The decision tree method achieves more successful results in terms of processing complete data than other statistical methods (Çalış, Kayapınar & Çetinyokuş 2014).

3.3.1. *Operating principle*

In a decision tree, data are classified from top to bottom, in other words, in a hierarchical form. A decision tree has three main components: root node, branch nodes and leaf nodes. Here, the leaf nodes demonstrate the final results of the classification. That is, leaf nodes themselves are not divided into other classes. Branch nodes are both the results of classification and are themselves classified into branch or leaf nodes. The root node is considered to be the primary object or data that needs to be classified. That is, it is not the result of any classification, and this is where the classification process initially begins (Figure 8) (Onan, 2015).

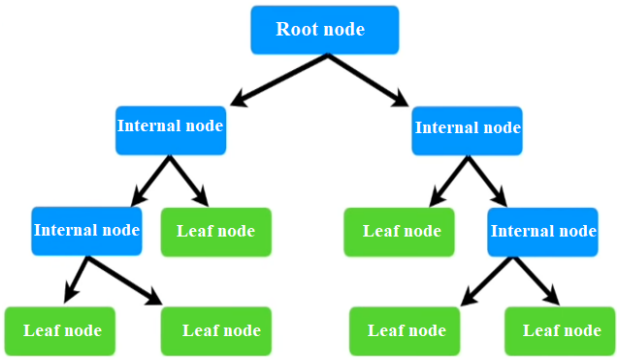

Figure 8. An example of a decision tree and its components

Algorithms such as ID3 (Iterative Dichotomiser 3), C4.5, Sliq (Supervised Learning in Quest), Sprint (scalable parallelizable induction of decision tree), CRT (classification and regression trees), CHAID (chi-square automatic interaction detection), REP (Reduced Error Prunning) Tree, Random Forest, Logistic Model Tree are used in construction of decision trees. These algorithms target minimization of classification error and indicators such as number of nodes and depth of tree (splitting) in order to build an optimal decision tree. Data is initially sent to one of these algorithms, then algorithms process this data and a decision tree is built. Built decision tree is applied to unclassified data and ensures finding classes for this data (Onan, 2015).

In a decision tree, various measurement criteria are used to determine which node is a root,

branch, or leaf node. These criteria vary according to the algorithms used to build the tree. For example, "Chi-square test" implemented by CHAID algorithm, Entropy or "Information gain" implemented by C4.5 algorithm, and "Gini impurity" criteria used by CRT algorithm. Gini impurity criterion is most widely used among the methods analyzed and listed in this article and its formula is expressed as follows (Çalış, Kayapınar & Çetinyokuş 2014; Yang, 2016):

$$Gini(D) = 1 - \sum_{i=1}^{k} P_i^2 \qquad (16)$$

where D- is the data set, k- number of general classes, P_i- occurring probability of created i^{th} class. "Gini impurity" value of data on each category is calculated in Figure 9. As "Good blood circulation" class has the least impurity value, it is located at the root node of the decision tree and other nodes are determined in accordance with this rule (Figure 9).

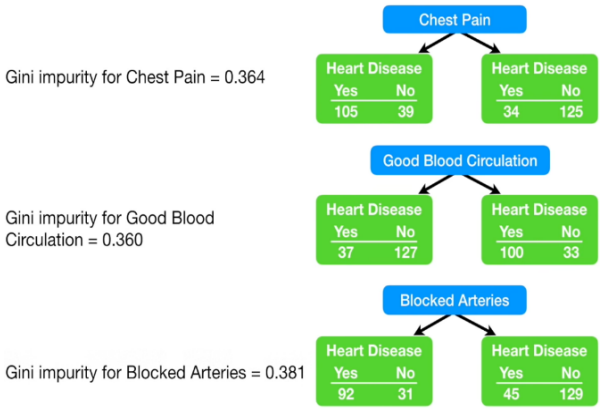

Figure 9. Classification of different categories of data in the decision tree based on "Gini impurity" criterion.

3.3.2. Decision tree algorithms

C4.5 algorithm was developed by Quinlan in 1993 and is one of the most widely used decision tree algorithms today. This algorithm uses "Entropy" criterion to assign classes in the decision tree. C4.5 is also an improved version of the ID3 algorithm developed by Quinlan. Thus, with the proposal of this algorithm, it became possible to use numerical data in decision trees. In addition, since the ID3 algorithm does not consider missing data in the dataset and this shortcoming was eliminated in the C4.5 algorithm. Another advantage of the C4.5 algorithm is that it can work with both discrete and continuous attributes (Çalış, Kayapinar & Çetinyokuş 2014).

CRT algorithm was developed by Breiman in 1984. In CRT algorithm classes are detected based on "Gini impurity" method. Algorithm can process both numerical and categorical data types (Çalış, Kayapinar & Çetinyokuş 2014).

CHAID algorithm is the most widely used decision tree algorithm after the CRT algorithm. Here, the classes are determined by the "Chi-square test" method (Çalış, Kayapinar & Çetinyokuş 2014).

3.3.3. Advantages and disadvantages

Main advantages of this method include the facility to understand, interpret, apply and integrate it into data base, its ability to work with both numerical and categorical data, fast and accurate obtaining of data (Onan, 2015; Çalış, Kayapinar & Çetinyokuş, 2014).

On the other hand, the main disadvantage is its instability. Thus, a small change in the data set can lead to a change in the overall structure of the tree. In addition, calculations become very complicated when some data are uncertain or if many results are interdependent. (Onan, 2015; Çalış, Kayapinar & Çetinyokuş, 2014).

4. Experimental results

In order to evaluate the studied clustering and classification algorithms in terms of factors such as performance, accuracy, volume and time, their application on the KDD CUP 99 dataset was studied and a comparative analysis was carried out based on obtained results.

KDD CUP 99 database was prepared by DARPA agency (Defense Advanced Research Projects Agency) of the United States Department of Defense. This data is used in order to detect anomalies in the network environment. Here, data is characterized as "network logs" (Tavallaee et al., 2009).

KDD CUP 99 database divides traffic into 5 classes (4 attack and 1 normal) and overall classes include 41 features. Attack classes are characterised as DoS (Denial of Services), probing, U2R and R2L. In addition, there are 4898430 log rows in the database about traffic. Of these, 972 780 were related to normal traffic and the rest to attack traffic. In total, these log lines included records of 22 records on attacks and 1 record on normal traffic (Nabiyev 2018; Tavallaee et al., 2009).

4.1. Comparative analysis

In order to evaluate the algorithms listed above, the experiments conducted on the basis of these algorithms and on KDD CUP 99 dataset were studied. Based on a series of studied experiments, each algorithm was comparatively analyzed based on various indicators (table 1, table 2).

Table 1. Comparative analysis of clustering algorithms on KDD CUP99 (10% part) data set

№	Algorithm	Number of records	Number of features	Number of classes	Performance time	Error percentage
1	KM	488735	23	3	70.7s	0.843%
2	CURE	488735	23	2	10.23s	0.5031%
3	EM	488735	23	3	1608.76s	0.8162%

Table 2. Comparative analysis of classification algorithms on KDD CUP 99 (10% part) data set

№	Algorithm	Number of records	Number of features	Number of classes	Performance time	Error percentage
1	KNN	494020	41	5	545.23s	11.9%
2	NB	494020	41	5	5.57s	11.68%
3	DT	494020	41	5	15.85s	7.94%

The number of records (rows) in Table 1 and Table 2 shows the number of log rows on traffic in KDD CUP 99 data set. Number of features means the number of features determining the measurement of records in dataset based on their characteristics. Performance time means performance period formed as a result of application of assigned calculation method, number of determined records and demonstrated algorithm.

Table 3. EM algorithm results on 3 clusters

Clusters	DoS1	DoS2	Normal
DoS1	107175	0	3493
DoS2	0	280327	463
Normal	33	0	97244

The error percentage parameter indicates the accuracy of grouping the data into appropriate clusters or classes according to their characteristics. Table 3 shows clustered data based on the EM algorithm as an example of error percentage calculation.

As seen from Table 3, several data were grouped in incorrect clusters. That is, here 3494 records in DoS1 cluster (normal), 463 records in DoS2 cluster (normal) and 33 records (DoS1) in Normal cluster were wrongly clustered. Error percentage is calculated based on these exact data.

$$XF = \frac{3493+463+33}{488735} \times 100\% = 0.8162\% \qquad (17)$$

In formula (16) the ratio of wrongly clustered records to total records according to the EM algorithm is shown, and as it can be seen, the obtained result corresponds to the error percentage indicator shown in table 1. Here, EP- stands for error percentage parameter.

5. Conclusion

In this article, a number of machine learning methods were studied and their comparative analysis was carried out in order to optimize the process of intellectual analysis of log files. The analysis process was performed in accordance with experiments conducted on the basis of several clustering and classification algorithms applied on the KDD CUP 99 dataset. At the end of conducted research and comparative analysis, it was determined that the CURE algorithm with a performance time of 10.23 seconds and an error percentage of 0.5031% showed the optimal result among the clustering algorithms, and the Naive Bayes algorithm showed a performance time of 5.57 seconds and an error percentage of 11.68% among the classification algorithms. In the next article, it is planned to inspect all the methods on the same data set and examine the differences in the results of these methods.

References

Alguliyev R.M., Aliguliyev R.M., Sukhostat, L.V. (2020). Parallel Batch k-means for Big Data Clustering. Computers & Industrial Engineering, vol 152. Parallel batch k-means for Big data clustering - ScienceDirect

Alıquliyev, R.M., Niftaliyeva, G.Y (2016). Application opportunities of Data Mining Technologies in E-government system analysis "Big Data: capabilities, multidisciplinary problems and perspectives" I Republican scientific-practical conference - Baku, 2016. - pp. 81-84. (in Azerbaijani)

https://ict.az/uploads/konfrans/biq_data/1-21_Gunay_Nifteliyeva_-_E-dovlt_sisteminin_analizind_data_mining_texnologiyalarn_ttbiq_imkanlar.pdf

Altunkaynak A., Başakın E.E. ve Kartal E., (2020). Air Polution Prediction with Wavelet K-Nearest Neighbour Method. https://dergipark.org.tr/en/download/article-file/1342958

Aslı Çalış, Sema Kayapınar, Tahsin Çetinyokuş (2014). An Application On Computer And Internet Security With Decision Tree Algorıthms In Data Mining. Journal Of Industrial Engineering Vol: 25 №: 3-4 P: (2-19) https://dergipark.org.tr/tr/download/article-file/752270

Aytuğ ONAN (2015). Comparative Performance Analysis of Decision Tree Algorithms in the Corporate Bankruptcy Prediction. Information Technologies Journal, Vol: 8, №: 1, https://dergipark.org.tr/en/download/article-file/75347

Babak R. Nabiyev (2018). Application of clustering methods network traffic for detecting DDoS attacks. Problems of Information Technologies, 2018, №1, 110–120. APPLICATION_OF_CLUSTERING_METHODS_NETWORK_TRAFFIC_FOR_DETECTING_DDOS_ATTACKS_azerb.pdf (jpit.az)

Babak R. Nabiyev, (2015). Network traffic clustering method. II Republican scientific-practical conference on multidisciplinary problems of information security, dedicated to the 150th anniversary of the International Telecommunication Union. - S. 213-215. https://ict.az/uploads/konfrans/2_konfrans/58.pdf

Burak D.B. (2020). Algorithm: Naive Bayes classifier. https://www.datasciencearth.com/algorithm-naive-bayes-classifier/

Edy Umargono, Jatmiko Endro Suseno, S.K. Vincesius Gunawan (2019). K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula. The 2nd International Seminar on Science and Technology (ISSTEC 2019) (PDF) K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula (researchgate.net)

Fargana J. Abdullayeva, Sabira S. Ojagverdiyeva (2021). An approach to identify vulgarism based on machine learning. Problems of Information Technologies, 2021, №2, 89–98. AN_APPROACH_TO_IDENTIFY_VULGARISM_BASED_ON_MACHINE_LEARNING.pdf (jpit.az)

Han J., Kamber M., Pei J., (2011). Data Mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann 744 p. https://www.sciencedirect.com/book/9780123814791/data-mining-concepts-and-techniques

Jingzhong Wang, Xia Li, (2010). An improved KNN algorithm for text classification. International Conference on Information, Networking and Automation (ICINA)., vol2, pp., 436-439. An improved KNN algorithm for text classification | IEEE Conference Publication | IEEE Xplore

Kesavaraj, G., Sukumaran, S. (2013). A Study On Classification Techniques in Data Mining. IEEE Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp 1–7. A study on classification techniques in data mining | IEEE Conference Publication | IEEE Xplore

Meral DEMİRALAY, A. Yılmaz ÇAMURCU, (2005). COMPARISON OF CLUSTERING CHARACTERISTICS OF CURE, AGNES AND K-MEANS ALGORITHMS. Istanbul Commerce University Journal of Science, 2005, №2, p.1-18. https://dergipark.org.tr/tr/download/article-file/199461

Muhammad Zulfadhilah, Yudi Prayudi, Imam Riadi (2016). Cyber Profiling using Log Analysis and K-Means Clustering A Case Study Higher Education in Indonesia. International Journal of Advanced Computer Science and Applications, vol 7. https://www.researchgate.net/publication/305737193_Cyber_Profiling_using_Log_Analysis_and_K-Means_Clustering_A_Case_Study_Higher_Education_in_Indonesia

Nadkarni, Prakash (2016). Clinical Research Computing and Core Technologies: Data Mining and "Big Data", pp., 187–204. Core Technologies: Data Mining and "Big Data" - ScienceDirect

Ömürbek, N., Dağ, O., Eren, H., (2020). Evaluation of Airports Clustered According to EM Algorithm by Side Count Method. Atatürk University Journal of Economics and Administrative Sciences, 34(2): 491-514. https://dergipark.org.tr/en/download/article-file/1049649

Pecht, Michael G., Kang, Myeongsu (2018). Machine Learning: Fundamentals. Machine Learning: Fundamentals - Prognostics and Health Management of Electronics - Wiley Online Library

Ramiz H. Shikhaliyev (2022). A method for Intelligent Scheduling Of Computer Networks Monitoring. Problems of Information Technology (2022), vol. 13, no. 1, 38-42. A_METHOD_FOR_INTELLIGENT_SCHEDULING_OF_COMPUTER_NETWORKS_MONITORING.pdf (jpit.az)

Sudipto Guha, Rajeev Rastogi, Kyuseok Shim (2001). CURE: AN EFFICIENT CLUSTERING ALGORITHM FOR LARGE DATABASES. Information Systems Vol. 26, No. 1, pp. 35-58. https://www.sciencedirect.com/science/article/abs/pii/S0306437901000084

Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set. IEEE Symposium on Computational Intelligencein Security and Defense Applications, 2009, pp.53–58.

Tim Zwietasch (2014). Detecting Anomalies in System Log Files using Machine Learning Techniques. (Detecting Anomalies in System log files using Machine Learning Techniques (d-nb.info))

Yadigar, N. Imamverdiyev, Babek R. Nabiyev (2014). Multi-classifier model for network traffic. Problems of Information Technologies, - 2014. - N: 2(6). - S. 68-74. https://jpit.az/uploads/article/az/MULTI_CLASSIFICATORY_MODEL_FOR_NETWORK_TRAFFIC_azerb._.pdf

Yang T. et al. (2016). Improve the Prediction Accuracy of Naive Bayes Classifier with Association Rule Mining. IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance, and Smart Computing, IEEE International Conference on Intelligent Data and Security, pp. 129-133. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.38 https://ieeexplore.ieee.org/document/5356528

Ying, S., Wang, B., Wang, L., Li, Q., Zhao, Y., Shang, J., Geng, J. (2021). An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labeled Samples. ACM Transactions on Knowledge Discovery from Data, 15(3), 1–22. An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labeled Samples | ACM Transactions on Knowledge Discovery from Data

Banu Zafer (2006). Unobservable class analysis and application. Yıldız Technical University, Graduate School of Natural and Applied Sciences, 2006 http://dspace.yildiz.edu.tr/xmlui/bitstream/handle/1/4189/0028352.pdf?sequence=1&isAllowed=y

Zhang, S., Xuelong L., Zong M., Xiaofeng Z., Cheng D., (2017). Learning k for kNN Classification. ACM Transactions on Intelligent Systems and Technology, vol 8, pp., 1–19. Learning k for kNN Classification | ACM Transactions on Intelligent Systems and Technology