

Development of a Method for Increasing the Efficiency of Identifying the State of a Computer Network by Selecting the Most Informative Features in the Output Data

Svitlana Gavrylenko and Vlad Zozulia

National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova str., Kharkiv, 61002, Ukraine

Abstract

The object of the study is the process of identifying the state of the computer network. The subject of the study is methods of selecting the most informative features in the initial data. The purpose of the work is to improve the efficiency of identification of the state of the computer network. Methods used: methods of artificial intelligence, machine learning, genetic and natural algorithms. The input data sets UNSW-NB 15, KDDCUP99, Kyoto2006, NSL-KDD, NSL-KDD, CIC-IDS-2017 DoHBrw-2020, which contain information about the normal functioning of the network and during intrusions, were used as input data. Software models based on genetic and natural algorithms were developed and researched: Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Cuckoo Search Algorithm, Bat Algorithm and Flower Pollination Algorithm for selecting the most informative features in the raw data at the stage of data preprocessing. To assess the quality of data preprocessing, a computer network state identification model based on the Random Forest algorithm was developed. The following results were obtained. The use of the above algorithms made it possible to significantly reduce the number of features of the complete dataset, reduce its size and speed up the training time of the model. At the same time, the best results were obtained when using the genetic algorithm, which made it possible to increase the learning speed of the models by 47% on average. Conclusions. According to the results of the study, the method of increasing the efficiency of identification of the state of the computer network by using genetic methods to select the most informative features in the source data was further developed.

Keywords

intrusion detection systems, computer networks, machine learning, feature informativeness, genetic and natural algorithms

1. Introduction

With the increasing dependence of the modern world on information technologies, issues of cyber security are becoming more relevant and important. In this environment, ensuring the security of networks and data takes on the highest priority. One of the key aspects in the field of cyber security is intrusion detection [1], the purpose of which is to detect illegal and malicious actions in computer systems and networks. Effective intrusion detection systems can prevent significant threats, minimize risks and protect valuable resources.

With the advent of large volumes of data and sophisticated attacks, the issues of efficiency and accuracy of intrusion detection systems become particularly important. In this context, the selection of the most informative features when building an intrusion detection model becomes a decisive factor for achieving optimal results. Reducing the dimensionality of data, while preserving relevant information, can significantly increase the effectiveness of intrusion detection and optimize the learning process.

The object of the study is the process of identifying the state of the computer network.

The subject of the study is methods of selecting the most informative features in the initial data.

Profit AI 2023: 3rd International Workshop of IT-professionals on Artificial Intelligence (Profit AI 2023), November 20–22, 2023, Waterloo, Canada


✉ svitlana.gavrylenko@khp.edu.ua (S. Gavrylenko); zozuliavlad@gmail.com (V. Zozulia)

ORCID [0000-0002-6919-0055](https://orcid.org/0000-0002-6919-0055) (S. Gavrylenko); [0000-0002-2168-3029](https://orcid.org/0000-0002-2168-3029) (V. Zozulia)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The purpose of the work is to improve the efficiency of identification of the state of the computer network.

2. Statement of the problem and relevance of the task

The presence of uninformative features in the data set can lead to several problems when building a model:

1. Increase in size. Uninformative features increase the size of the data, which can make analysis and modeling tasks more difficult. This can lead to an increase in the computational complexity and training time of the models:
2. Increased noise. Noninformative features add "noise" to the data because they do not contain useful information about data dependencies. This can reduce the ability of models to identify true patterns and make them less robust to changes in the data.
3. Deterioration of model performance. Incorporating uninformative features into a model can degrade its performance. The model can overlearn by analyzing uninformative features and have a weak generalization ability to new data.
4. Increasing data requirements. Having uninformative features means that more data may be needed to train the model so that the model can have useful dependencies. This can be expensive and time consuming.
5. Interpretation difficulties. Non-informative features can complicate model interpretation and make it difficult to understand the influence of specific features on results.

To solve these problems, feature selection methods are often used, which allow identifying and removing uninformative features from the data set, namely: feature importance analysis method (Feature Importance Analysis [2]), recursive feature elimination (Recursive Feature Elimination [3]), wrapper methods, genetic and natural algorithms, etc. These techniques can help identify and remove uninformative features, improving model performance and facilitating data analysis.

However, feature selection may lead to the loss of some information, and this may reduce the performance of the model. Feature selection methods can be sensitive to data variability. In some cases, features that seem uninformative may be useful under other conditions. In addition, many methods, for example, filter methods, mutual information methods, recursive feature elimination (RFE), etc. consider signs independently of each other and do not take into account their interaction.

Although there are many different feature selection methods, choosing the appropriate method can be complex and depends on the specific machine learning method. Usually, it is necessary to investigate several methods and evaluate their impact on model performance in order to choose the best approach for feature selection. Thus, improving the development of methods for selecting informative features in order to improve the quality of the model is an urgent task and requires research.

3. Review of scientific publications

The use of genetic and natural algorithms was investigated in this work to evaluate the informativeness and selection of model output data by creating optimal (or appropriate) subsets of features from a set of available features. The choice of genetic algorithms is justified by their ability to take into account the interaction between features and evaluate how combinations of features affect the predictive ability of the model, which is important when detecting intrusions into computer networks.

In this work, to evaluate the informativeness of features, search and their influence on the efficiency of intrusion detection systems, the use of genetic algorithms is investigated: genetic algorithm (Genetic Algorithm, GA [4]), particle swarm optimization (PSO [5]), algorithm of

differential of evolution (Differential Evolution, DE [6]), Cuckoo Search Algorithm (CSA [7]), Bat Algorithm (BA [8]), Flower Pollination Algorithm (FPA [9]).

A Genetic Algorithm is a type of calculation that solves optimization tasks and is based on the methods of natural evolution: inheritance, crossing, mutation, and selection. A distinctive feature of the genetic algorithm is the emphasis on the use of the crossover operator, which performs the operation of recombining candidate solutions, the role of which is similar to the role of crossover in living nature.

In the simplest case, the optimization task consists in finding the extremum (minimum or maximum) of the objective function by systematically sorting input values from a given set and calculating its value

$$Y = f(x_1, x_2, \dots, x_n) \quad (1)$$

where Y is the objective function that depends on the parameters (x_1, x_2, \dots, x_n) and, depending on the problem, tends to the maximum or minimum value.

When using a genetic algorithm, the selection of traits or "trait selection" can be described as follows:

An example of numbered list is as following.

1. Initialization of the population or selection of the initial population of chromosomes.
2. Assessment of the fitness of each individual in the population based on some criterion, which can be, for example, the accuracy of the model on test data. The better the individual is suited to the task, the higher his adaptability.
3. Selection of individuals to create a new population, taking into account their adaptability. Individuals with higher fitness have a greater chance of selection.
4. Crossing over and mutation, for example combining traits from two parents, and mutation may involve the random addition or deletion of traits.
5. Replacement of the previous generation with a new one. The process of selection, crossing, mutation and assessment of fitness is repeated for each generation.
6. Completion of the algorithm when a certain number of generations is obtained or before a stopping condition is met, such as reaching the desired accuracy or a time limit. The obtained subset of features is optimal and can be used to train the model in order to reduce the dimensionality of the data, improve the performance of the model and reduce the risk of overtraining.

The Particle Swarm Algorithm is an optimization algorithm inspired by the behavior of swarms of particles in nature, such as birds, fish, or insects. It is used to solve optimization problems where you need to find an adaptive value function in a multidimensional space by exploring it using a configuration of "particles" that move through space in search of a better solution. PSO optimizes a function by maintaining a population of possible solutions, swarm particles, moving these particles in the solution space according to a simple formula. Movements are subject to the principle of the best position found in this position, which changes when finding fractions of more favorable positions.

The algorithm of Differential Evolution is a method of multidimensional mathematical optimization that belongs to the class of stochastic optimization algorithms (that is, it works with the use of random numbers). It simulates the main evolutionary processes in living nature: crossing, mutation, selection. The method is intended for finding the global minimum (or maximum) of undifferentiated nonlinear, multimodal (may have a large number of local extrema) functions from many variables. The method is easy to implement and use (contains few control parameters that require selection), is easily parallelized.

The differential evolution algorithm is an optimization method designed to solve unconstrained optimization problems. It is based on the ideas of evolution and is used to find the global optimum in multidimensional spaces. This is how the differential evolution algorithm works:

The Cuckoo Search Algorithm is a metaheuristic for optimization problems that simulates the behavior of cuckoo birds forced to use an aggressive breeding strategy. It works by dividing the search space into niches (nests), where each niche represents a potential solution to the optimization problem and the cuckoo egg is a new solution.

The Bat Algorithm is a metaheuristic global optimization algorithm. It simulates the process of searching for food by bats, taking into account their ability to echolocate with different pulse rates and volumes.

In the basic bat algorithm, each bat is treated as a "massless and size free" particle representing a feasible solution in the solution space. For different fitness functions, each bat has a corresponding function value and determines the current optimal individual by comparing the function values. Then, the acoustic wave frequency, velocity, pulse emission rate, and loudness of each bat in the population are updated, the iterative evolution continues, the current optimal solution is approximated and generated, and finally the global optimal solution is generated. The algorithm updates the frequency, speed and position of each bat.

The standard algorithm requires five basic parameters: frequency, loudness, ripple, and loudness and ripple coefficients. Frequency is used to balance the effect of the optimal historical position on the current position. An individual bat will search far from the group's historical position when the search frequency range is large, and vice versa.

The Flower Pollination Algorithm is a highly efficient metaheuristic optimization algorithm inspired by the pollination process of flower species. It is aimed at predicting the movement and interaction of pollen grains and other particles inside a flower.

4. Output data

Six different datasets covering various scenarios of network activity and types of attacks were used as raw data:

1. The DoHBrw-2020 dataset [10] is focused on the analysis of network activity related to the DNS over HTTPS (DoH) protocol. The data was obtained from the result of the analysis of the traffic associated with the use of the DoH protocol. DoHBrw-2020 provides data for analysis of attacks masquerading as encrypted traffic.
2. The UNSW-NB15 dataset [11] was developed as a comprehensive dataset for evaluating network intrusion detection. It contains different types of attacks and normal network activity. Data were collected in the university network over a period of five months. Network activity logs are included, and artificial attacks are created to cover scenarios more fully. UNSW-NB15 contains over 2 million records and covers attacks of varying complexity.
3. The KDDCUP99 dataset [12] was prepared for the KDD Cup 1999 intrusion detection competition. It contains different types of attacks and normal activity. The data was collected in a real network environment as part of the KDD Cup 1999 and represents a record of network activity, including various types of attacks. kddcup99 has an uneven distribution of classes and problems with re-presentation of some attacks.
4. The Kyoto2006 dataset [13] specializes in the analysis of DoS type attacks and forged packets. It includes records of network activity associated with such attacks. The data was collected in a real network environment using special sensors to detect DoS attacks and packet spoofing. Kyoto2006 provides unique data for packet-level attack analysis.
5. The NSL-KDD dataset [13] was created on the basis of the kddcup99 dataset in order to eliminate shortcomings and add other types of attacks. NSL-KDD includes attacks from the original kddcup99 dataset, as well as added artificial attacks and normal activity. NSL-KDD provides a more even distribution of classes and a variety of attack types.
6. The CIC-IDS-2017 dataset [14] is designed to detect modern attacks, including those targeting applications. The data was collected in a real network environment and includes a variety of attacks from both traditional and emerging threat types. CIC-IDS-2017 provides up-to-date data for analysis and detection of modern threats.

Each of these datasets provides unique opportunities to investigate and analyze the performance of genetic algorithms for selecting informative features in the context of intrusion detection.

5. Experimental part

This section presents the results of experiments using six different genetic algorithms for extracting informative features in the tasks of detecting intrusions into cyber security systems.

Developed feature analysis software in Python in Jupiter Notebook environment. The following methods of character analysis were studied: Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Cuckoo Search Algorithm, Bat Algorithm, Flower Pollination Algorithm.

The Random Forest method [15,16] was used as the basic data classification model.

The experiments were carried out by sorting through the aforementioned datasets and genetic and natural algorithms. Each experiment included the following steps:

1. Selection of a set of input data (full dataset).
2. Selection of the basic algorithm and model training, evaluation of the informativeness of the features of the complete dataset.
3. Removal of non-informative features determined by the algorithm in the previous step and formation of a shortened dataset.
4. Estimation of the size of the full and reduced dataset.
5. Training of a classifier based on the Random Forest algorithm using full and reduced datasets.
6. Assessment of classification accuracy on full and reduced datasets.
7. Estimation of classifier training time on full and reduced datasets.

According to the results of the study, the following quality indicators of the model were obtained (Figure 1).

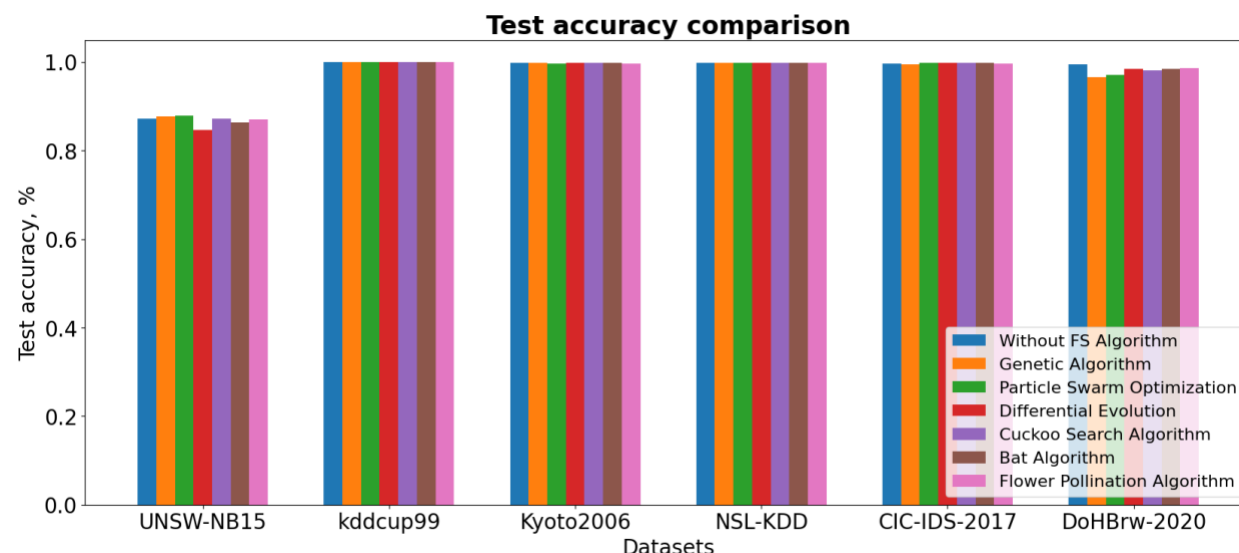


Figure 1: Comparison of model accuracy when using different datasets and different genetic algorithms for selecting the most informative features

As can be seen from Fig. 1, the use of a reduced dataset as input data of the model did not lead to a significant deterioration of its accuracy in comparison with the full dataset. For the UNSW-NB15 data set, the use of the genetic algorithm (GA) and the particle swarm algorithm (PSO), on the contrary, led to a slight increase in the accuracy of the model.

The use of genetic algorithms made it possible to reduce the number of features of the complete dataset by an average of 65% (Figure 2), which made it possible to reduce its size from two to six times (Figure 3).

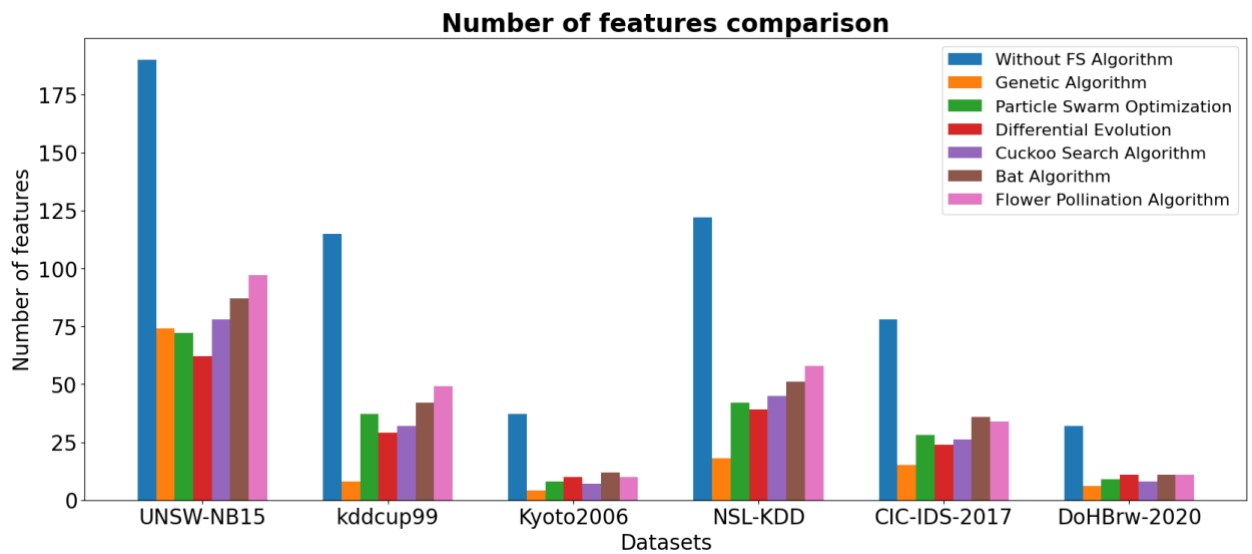


Figure 2: Comparison of the number of the most informative traits when using different genetic selection algorithms

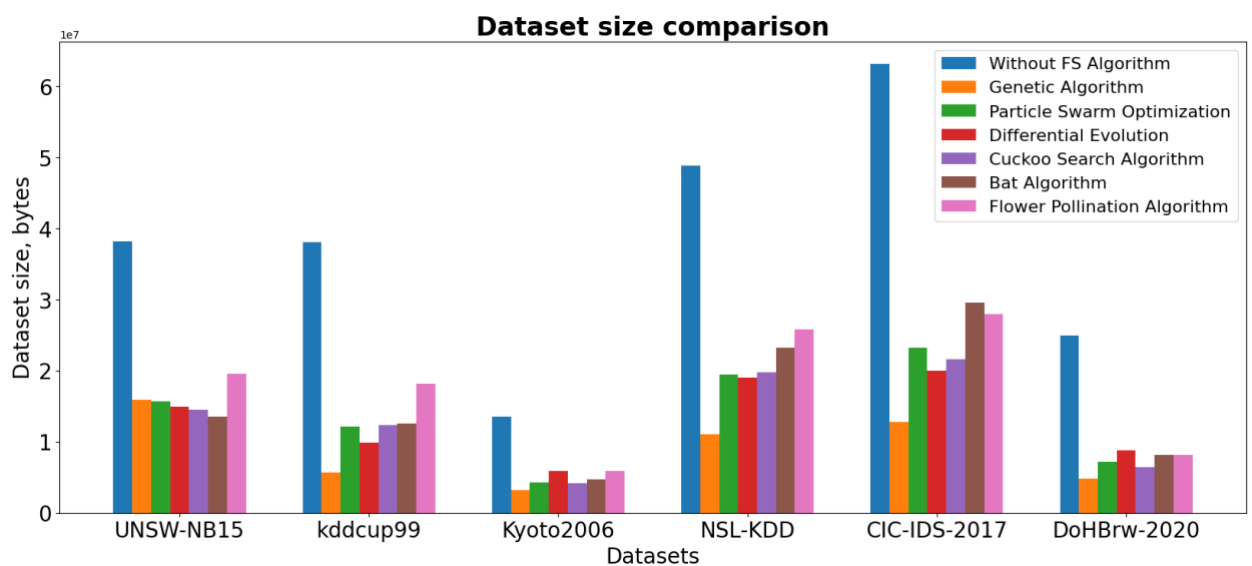


Figure 3: Comparison of the size of datasets when using different genetic algorithms for selecting the most informative features

Decreasing the size of the dataset led to a decrease in the training time of the model. A comparative analysis of the increase in the learning speed of the model in percentage terms when using different datasets and different genetic algorithms for selecting the most informative features is given in the table. 1. As can be seen from the table. 1, different algorithms have different effects on increasing the speed of model training when using different datasets. However, the best results were obtained when using the genetic algorithm. The training speed of models when using different datasets decreased from 36.36% to 59.38% and, on average, set 47%. At the same time, the classification accuracy remained at the level of about 99%.

Table 1.

Comparative analysis of the increase in the learning rate of the model (%) when using different datasets and different genetic algorithms for selecting the most informative features

Algorithm	UNSW-NB15	KDDCUP99	Kyoto2006	NSL-KDD	CIC-IDS-2017	DoHBrw-2020
GA	50,00	50,00	50,00	38,46	36,36	59,38
PSO	42,86	33,33	50,00	38,46	45,45	37,50
DE	50,00	33,33	33,33	38,46	54,55	34,38
CS	50,00	33,33	50,00	30,77	40,91	59,38
BA	42,86	33,33	50,00	30,77	31,82	46,88
FPA	35,71	16,67	33,33	30,77	45,45	53,13

6. Conclusions

In this work, the use of various genetic and natural algorithms for the selection of the most informative features in order to increase the efficiency of identification of the state of the computer network is investigated at the stage of data preprocessing. Considered: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Cuckoo Search Algorithm (CSA), Bat Algorithm (BA), Flower Pollination Algorithm (FPA). Their software models were developed in the Jupiter Notebook environment using Python. To assess the quality of data preprocessing, a computer network state identification model based on the Random Forest algorithm was developed.

The following sets (datasets) were used as source data: UNSW-NB 15, KDDCUP99, KYOTO2006, NSL-KDD, NSL-KDD, CIC-IDS-2017 DoHBrw-2020, which contain information about the normal functioning of the network and during intrusions.

It was found that the use of genetic algorithms made it possible to significantly reduce the number of features of the complete dataset and reduce its size to 63%. Reducing the size of the dataset accelerated the training time of the model by up to 59%. At the same time, the accuracy of the model did not decrease significantly.

Studies have shown that the choice of genetic or natural algorithm type depends on the input data. In our study, better results were obtained when using a genetic algorithm, which made it possible to increase the learning rate of models by 47% on average.

According to the results of the study, the method of increasing the efficiency of identifying the state of the computer network by using the procedure for selecting the most informative features in the source data based on the genetic algorithm was further developed.

Thus, the obtained results testify to the effectiveness of using genetic algorithms for the selection of informative features at the stage of data pre-processing in the tasks of detecting intrusions into cyber security systems. Accelerating model training and reducing the amount of data can significantly improve the performance of real-time systems, which is an important direction for further research.

References

- [1] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, Farhan Ahmad, Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*. (2021) 32:e4150. doi: 0.1002/ett.4150
- [2] Bouchlaghem Younes, Yassine Akhiat, Souad Amjad Feature Selection: A Review and Comparative Study. *E3S Web of Conferences* (2022) 351(1):01046. doi: 10.1051/e3sconf/202235101046.

- [3] Gbashi Ekhlas, Mohammed, Bilal, Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination, *Engineering and Technology Journal*, 39(7), (2021). doi:10.30684/etj.v39i7.1695.
- [4] Anita Thengade, Rucha Dondal, Genetic Algorithm – Survey Paper, *IJCA Proc National Conference on Recent Trends in Computing*, NCRTC. 5, (2012) 25-29.
- [5] Eberhart Shi Yuhui, Particle swarm optimization: Development, applications and resources. *Proceedings of the IEEE Conference on Evolutionary Computation*, ICEC, 1 (2001) 81 – 86. doi:10.1109/CEC.2001.934374.
- [6] Das Swagatam , Suganthan Ponnuthurai, Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evolutionary Computation*, 15 (2011) 4-31.
- [7] Amir Gandomi, Xin-She Yang, Amir Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering With Computers*, 29 (2013) 245-245. doi:10.1007/s00366-012-0308-4.
- [8] Xin-She Yang, Bat Algorithm: Literature Review and Applications. *International Journal of Bio-Inspired Computation*, 5. (2013).141-149. doi:10.1504/IJBIC.2013.055093.
- [9] Yang, XS. (2012). Flower Pollination Algorithm for Global Optimization. In: Durand-Lose, J., Jonoska, N. (eds) *Unconventional Computation and Natural Computation*. UCNC 2012. *Lecture Notes in Computer Science*, vol 7445. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-32894-7_27.
- [10] Jafar Mousa, Al-Fawa'reh Mohammad, Jafar Shifa, Analysis and Investigation of Malicious DNS Queries Using CIRA-CIC-DoHBrw-2020 Dataset 2. *Manchester Journal of Artificial Intelligence and Applied Sciences(MJAIAS)*, (2021) 65-70.
- [11] Moustafa Nour, Slay Jill. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) (2015). doi:10.1109/MilCIS.2015.7348942.
- [12] Tavallaee Mahbod, Bagheri Ebrahim, Lu Wei, Ghorbani Ali, A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium. Computational Intelligence for Security and Defense Applications*, CISDA. 2 (2009). doi:10.1109/CISDA.2009.5356528.
- [13] Protic, Danijela. Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets. *Vojnotehnicki glasnik*. 66 (2018) 580-596. doi:10.5937/vojtehg66-16670.
- [14] Jose Jinsi, Jose Deepa, Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset, *International Journal of Electrical and Computer Engineering (IJECE)* 13 (2023). 1134-1141. doi:10.11591/ijece.v13i1.pp1134-1141.
- [15] Ali Jehad, Khan Rehanullah, AhmadNasir, Maqsood, Imran, Random Forests and Decision Trees, *International Journal of Computer Science Issues(IJCSI)*, 9 (2012) 272-278
- [16] S. Gavrylenko, V. Chelak and O. Hornostal, "Ensemble Approach Based on Bagging and Boosting for Identification the Computer System State," *XXXI International Scientific Symposium Metrology and Metrology Assurance (MMA)*, Bulgaria, (2021) 1-7, doi: 10.1109/MMA52675.2021.9610949.