International Journal of Computer Applications Technology and Research
Volume 6–Issue 11, 461-466, 2017, ISSN:-2319–8656

# Intrusion Detection System Using Genetic Algorithm and Data Mining Techniques Based on the Reduction Features

Mohammad Ghalehgolabi
Electronic and Systems Engineering
Department of Rahjuyan Danesh Borazjan
Bushehr, Iran

Amin Rezaeipanah
Faculty of Computer
Department of Rahjuyan Danesh Borazjan
Bushehr, Iran

**Abstract**: An intrusion detection system is the process for identifying attacks on network. Choosing effective and key features for intrusion detection is a very important topic in information security. The purpose of this study is to identify important features in building an intrusion detection system such that they are computationally effcient and effective. To improve the performance of intrusion detection system, this paper proposes an intrusion detection system that its features are optimally selected using genetic algorithm optimization. The proposed method is easily implemented and has a low computational complexity due to use of a simplified feature set for the classification. The extensive experimental results on the NSL-KDD intrusion detection benchmark data set demonstrate that the proposed method outperforms previous approaches, providing higher accuracy in detecting intrusion attempts and lower false alarm with reduced number of features.

**Keywords**: intrusion detection; genetic algorithm; distribution function; NSL-KDD; feature selection.

## 1. INTRODUCTION

In recent year, due to the growing use of smart devices and the Internet, network traffic is rapidly increasing. A Cisco report found the following : "Global IP traffic in 2012 stands at 43.6 exabytes per month and will grow threefold by 2017, to reach 120.6 exabytes per month" [1]. Intrusions are defined as attempts or action to compromise the confidentiality, integrity or availability of computer or network [2]. Intrusion detection systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems [3]. Feature Selection (FS) is the process of removing features from the original data set that are irrelevant with respect to the task that is to be performed. So not only the execution time of the classifier that processes the data reduces but also accuracy increases because irrelevant or redundant features can include noisy data affecting the classification accuracy negatively [4]. In this paper, we suggest a new feature selection method that uses the features distribution function. The decision tree [5] and k-nearest neighbor [6] classifiers will be evaluated with the NSL-KDD dataset to detect attacks on four attack categories: Dos, Probe, R2L, and U2R. The decision tree classifier's results are computed for comparison of feature reduction methods to show that our proposed model is more efficient for network intrusion detection.

The remainder of the paper is organized as follows: Section 2 give an overview of feature selection methods and intrusion detection. The basic concept of the proposed method are presented in Sections 3 and the experimental results are presented in Section 4. Finally the paper is concludes with their future work in section 5.

## 2. RELATED WORKS

Intrusion detection techniques using data mining have attracted more and more interests in recent years. Feature selection is important to improving the efficiency of data mining algorithms [7]. Different researchers propose different algorithms in different categories, from Bayesian approaches [8] to decision trees [9], from rule based models [10] to functions studying [11]. The detection efficiencies therefore are becoming better and better than ever before. In recent years, researchers turn their focus on heuristic and hyper-heuristic methods for features selection. Several examples on these methods including Genetic Algorithm [12], Particle Swarm Optimization [13], and Ant Colony Optimization [14].

Sung and Mukkamala proposed a well-known closedloop FS method for SVM-based IDS, called SVM-RFE, which recursively eliminated one feature at a time and compared the resulting performance in each SVM test [15]. They also ranked six significant features [16]. Intrusion Detection in NEAR System by Anti-denoising Traffic Data Series using Discrete Wavelet Transform was presented by Vancea [17]. In [18] uses NGSA-II for wrapper-based feature selection and GHSOM-pr as the classifier to build efficient IDS. D. Sequeira [19] discussed in their research different types of firewalls. Traditional firewalls cannot detect internal attacks such as flooding attacks, user-to-root attacks, and port scanning because they only sniff out network packets at the network boundaries. Moreover, traditional firewalls cannot differentiate between ordinary traffic and DoS attack traffic, as mentioned by [20]. Warsi et al. [21] present a selective iteration based particle swarm optimization (SIPSO) for intrusion detection system with an upgraded beginning masses and decision director, to capably distinguish diverse sorts of interferences. Aghdam and Kabiri considered the feature selection using ant colony optimization in detecting the attacks [15]. The purpose of this study is to identify important features in building an intrusion detection system such that they are computationally effcient and effective.

## 3. PROPOSED SYSTEM

Some data sets like NSL-KDD have a lot of features. On the other hand, all of these features do not play a positive role in data categorization. Therefore, you need to select a subset of the best features. In this research, a genetic algorithm is used

to select the desired features. This method operates on the basis of the features distribution function analysis. This factor helps to improve the genetic algorithm chromosomes by recognizing the peculiarities. The proposed method can work on a dataset of different dimensions. To evaluate the selected features, two well-known data mining techniques, decision tree (DT) and k-nearest neighbor (KNN) are used. Figure 1 shows the flowchart of the proposed method.

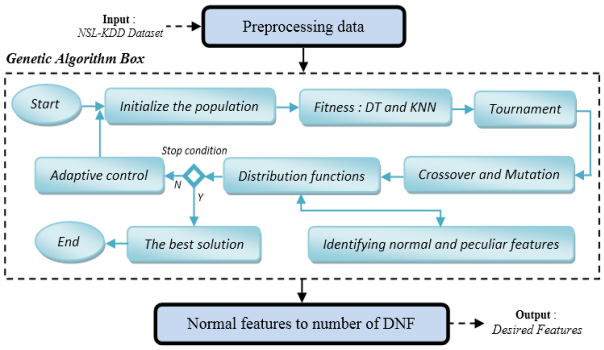


Figure. 1 Overall process of proposed intrusion detection system

## 3.1 Preprocessing Data

The first step in the creation of any model based on data mining techniques is the preprocessing of data. Pre-processing is done to prepare data for processing as well as improve the quality of real data. This step involves converting string-to-number properties, normalizing and disassembling data.

## 3.2 Improved Genetic Algorithm for Features Selection

Genetic algorithm was introduced by Holland in 1970, inspired by genetics and Darwinian evolution theory [22]. In this research, the structure of the chromosome is considered with regard to the number of each attribute. Each chromosome is a string of bits with values of 0 and 1 with a length of the total number of features. The genes of a chromosome show the desirable features that will be involved in the classification of the data. In this research, the number of desirable features (DNFs) is fixed in terms of test and error. In the proposed method, the genetic algorithm is implemented sequentially, so each repetition requires the production of a primary population of features. The genetic algorithm begins with an initial population of chromosomes randomly. Then the cluttered and cluttered features of the search space are extracted and used to generate the population in later stages. Compact features are a vector of attributes that are used in the production of the population. Pecularities are vector of features that their use in pre-population generation does not have desirable results and will not be used in the production of new populations. The fitness criterion of chromosomes is the error rate of the classification of data. Because of the expeditious calculation of fitness, two classifiers of KNN and DT have been used.

The chromosome selection operator, the tournament, and the crossover operator was one-point cross over. In the one point cross over operator, single particle genes for parents are exchanged to create new members. After applying this operator to probability Cr, the number of 1 chromosome genes must be constant. A one-point crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children. An example of this operator is shown in Figure 2.

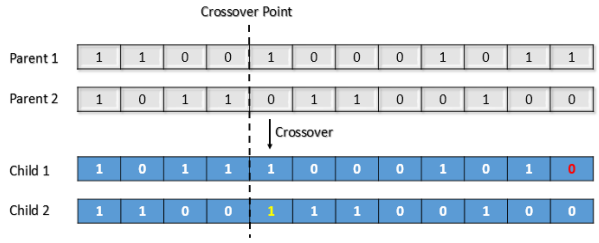


Figure. 2 One-point crossover operator suggested

In this example, features number is 12 and Desired Number of Features is 6, two genes of the children have changed. With applying the one-point crossover, the number of features in the first child is 7 and in the second child is 5. The number of genes in a chromosome should be equal to 6 in the offspring. So, in the first child we will random delete a gene and in the second child, we also random select a gene in unused features.

The mutation change of the bit is applied to one of the produced chromosomes. The role of the mutation in the genetic algorithm is to restore the genetic loss of the population, which provides access to all of the search space. The mutant operator is applied to the probability Mr for each gene. An example is shown in Figure 3.


Figure. 3 Bit change mutation operator suggested

The number of children created is equal to the number of parents. In order to determine the population of the next generation, the chromosomes of the population of the previous and current generations are sorted according to the fitness criterion in descending order. Then, the 25% elemental list (best ones) goes straight to the next generation. The endpoints of 25% chromosomes are removed (the worst ones) and finally the rest of the population are randomly selected from the remaining chromosomes.

One of the interesting phenomena of genetic algorithms is the production of intermediate-generation chromosomes that have a high degree of fitness. These chromosomes may be destroyed due to the application of mutant and crossover operators and no longer be produced.

In this research, elitism is used to preserve these chromosomes. In each generation, a chromosome with the best amount of fitness is transmitted directly to the next generation.

## 3.3 Using the Features Distribution Function in Identifying Normal and Peculiar Features

At the end of the genetic algorithm, a population of solutions is obtained. In most techniques, the features used in the best solution are considered as desirable features and classify educational data based on these features. The structure of the genetic algorithm is based on random search, which is why it does not always produce the same optimal solution. With these conditions, it will not be possible to find the desirable features that will best serve the classification of data.

Therefore, in this research, an approach has been proposed that largely leads to the selection of the best features. Our goal in this section is to identify the normal and peculiar features due to the outcomes of the genetic algorithm. To realize this goal, features distribution function (FD) has been used in the population. The distribution of any feature in the population indicates the degree of repetition of that feature.

Distribution of the characteristics of the population in the population is the rate of repetition of these features in parts of the population with high fitness. For example, the normal population are solutions that their fitness is greater than the overall fitness of the whole population. Also, the distribution of peculiar features is the frequency of these features in parts of the population with low fitness. For example, a rough population is a solution that is less than the overall fitness of the whole population.

The distribution function of a feature in a normal population is the ratio of its recurrence to the total population and the distribution function of an attribute in the peculiar population is the ratio of its recurrence to the entire peculiar population. Table 1, shows an example of the distribution function of the features.

In this example, according to the average population criteria, 4 solutions for normal population and 6 solutions for pecliar population were selected. The frequency of the first feature (F1) in the normal population is 3 and in the pecliar population is 2.

**Table 1. Example of the distribution function**

| Solution | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | Fitness |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 87 |
| **2** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 85 |
| **3** | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 78 |
| **4** | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 70 |
| **5** | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 50 |
| **6** | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 45 |
| **7** | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 40 |
| **8** | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 39 |
| **9** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 37 |
| **10** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 32 |
| **normal** | 3/4 | 2/4 | 2/4 | 4/4 | 1/4 | 2/4 | 0/4 | 2/4 | - |
| **peculiar** | 2/6 | 1/6 | 3/6 | 3/6 | 4/6 | 4/6 | 4/6 | 3/6 | - |

Therefore, the distribution function for this property is $FD_1^{normal} = 3/4$ for the normal population and $FD_1^{peculiar} = 2/6$ for the peculiar population. Due to the distribution function of the features, the list of normal and peculiar features are determined. In normal population, features with a distribution function higher than a constant value, such as α, are added to the list of normal properties.

Also, in peculiar population, features with a distribution function less than constant, such as β, are added to the list of pecliar properties. The parameters α and β control the similarity of the solutions (selection pressure) to select a feature in a normal and peculiar population. Given the number of desirable featres, the genetic algorithm is repeatedly repeated to find DNF of normal features. To help the genetic algorithm to find optimal solutions, a list of the normal and peculiar features is used to generate primary population. So that the initial population contains all the normal features and does not include any peculiar features. By fixing a number of features, this strategy significantly reduces the search space. Applying this limitation in the initial population will change the function of the two combinatory and mutation operators. Therefore, these operators should not add or remove features that violate the criterion of building primary population.

## 3.4 Adaptive Control of Parameters

Adaptive control of the parameters is in fact a method in the control theory in order to adapt the control system to the variable parameters in the system. The basis of comparative control is based on the estimation of the parameters. In this research, the values of the parameters Mr, α and β change during the implementation of the algorithm. The mutation rate parameter at the beginning of work has a relatively high value and decreases sequentially in the process of running the algorithm.The similarity parameter also initially contains a high percentage of the selected space, but it is reduced by repeating the algorithm and because of the difference between the selected features. The α and β parameters decrease by ε in the case of failure to improve the identification of the normal features in a constant number. This method partially solves the problem of the early integration of the genetic algorithm with constant rate operators. Relationships (1) and (2) are used for comparative control of two parameters of mutation rate and similarity.

$$C_r = \begin{cases} k_1 \dfrac{iter}{MaxIter} \times C_r & f' \geq \overline{f} \\ C_r & f' < \overline{f} \end{cases} \quad (1)$$

$$M_r = \begin{cases} k_2 \dfrac{iter}{MaxIter} \times M_r & f' \geq \overline{f} \\ M_r & f' < \overline{f} \end{cases} \quad (2)$$

Where k1, k2 < 1 are two constant values that control the deceleration of Cr and Mr. $\overline{f}$ and $f'$ are the population fitness average of the pre-generation and current generation population, respectively.

# 4. EXPERIMENTAL RESULTS

The NSL-KDD dataset was used to evaluated the performance of the proposed method [23]. This dataset contains 41 features and 5 classes (a normal class and 4 types of attack classes Dos, R2L, U2R and Probing). To implement the proposed method, the Matlab version 2016a software has been used. The results obtained from the experiments were used to increase the accuracy of the evaluation, a mean of 30 repetitions of the test.

In the implementation, the population size of 25, the number of generations 30, the rate of composition is 0.85 and the rate of mutation is 0.15. The pressure rate of the algorithm is considered in selecting the normal features α = 0.95 and the peculiar features β = 0.90. The number of desirable features selected according to the test and error were at best 23. Selected features of the proposed method for the NSL-KDD dataset are shown in Table 2.

Figure 4 and 5 shows the performance of two classifier of KNN and DT in terms of accuracy and Convergence speed on the chromosomes produced, Respectively.

**Table 2. Selected features of the proposed method**

| No. | attribute name | No. | attribute name |
|---|---|---|---|
| 1 | duration | 14 | root_shell |
| 2 | protocol_type | 16 | num_root |
| 3 | service | 17 | num_file_creations |
| 4 | flag | 18 | num_shells |
| 5 | src_bytes | 19 | num_access_files |
| 6 | dst_bytes | 25 | serror_rate |
| 7 | Land | 27 | rerror_rate |
| 8 | wrong_fragment | 29 | same_srv_rate |
| 11 | num_failed_logins | 30 | diff_srv_rate |
| 12 | logged_in | 34 | dst_host_same_srv_rate |
| 13 | num_compromised | 37 | dst_host_srv_diff_host_rate |
| 38 | dst_host_serror_rate | - | - |

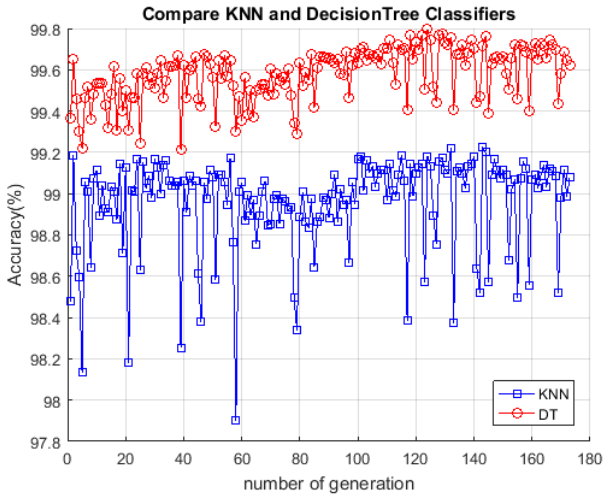


Figure 4. The performance of two classifier of KNN and DT in terms of accuracy on the chromosomes produced

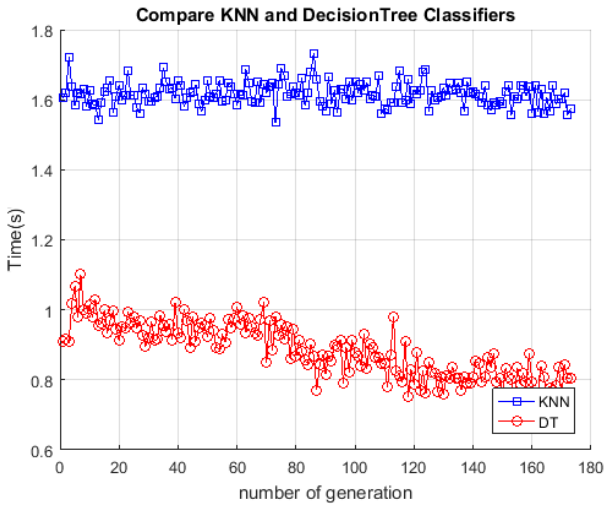


Figure 5. The performance of two classifier of KNN and DT in terms of Convergence speed on the chromosomes produced

The DT classifier method has a better performance than KNN and for this purpose the classification results are for comparison based on DT. In this research, the Accuracy, Precision, Recall and F-measure are used to evaluate the performance of the proposed method. The most important criterion for determining the efficiency of a classification model is Accuracy. This criterion calculates the precision of a single class, defined by relation (3).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

The FN, TN, FP, and TP parameters represent different states for the classes, which are False Negative, True Negative, False Positive, and True Positive, respectively. The precision criterion shows the precision of the class I classification with respect to all the items that have been proposed for the sample by the classifier. Equation (4) shows how this criterion is calculated.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (4)$$

The Recall criterion shows the accuracy of the class i classification for all samples with the i label. This criterion is calculated by equation (5).

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (5)$$

The F-measure criterion is calculated from the combination of two precision and recall criteria according to equation (6). This criterion is used in cases where it is not possible to attach special importance to each of the two criteria of Precision and Recall.

$$F - measure = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (6)$$

The effectiveness of intrusion detection systems can be assessed by the proposed criteria. The collision matrix of the Intrusion Detection System data is calculated for each of the four classes of attacks along with the normal class and is shown in Table 3. The table lists the number of records for each attack with the number of predictions.

**Table 3. The collision matrix is divided by type of attack**

| Actual Records | | | Predicted | | | | |
|---|---|---|---|---|---|---|---|
| Records Type | | Number | Normal | DOS | U2R | R2L | Probe |
| Normal | Train | 67343 | 67303 | 6 | 7 | 11 | 16 |
| | Test | 9710 | 9683 | 3 | 2 | 7 | 1 |
| DOS | Train | 45927 | 8 | 45909 | 0 | 0 | 10 |
| | Test | 7458 | 1 | 7454 | 1 | 0 | 0 |
| U2R | Train | 52 | 7 | 0 | 44 | 1 | 0 |
| | Test | 200 | 2 | 0 | 197 | 3 | 0 |
| R2L | Train | 995 | 13 | 0 | 1 | 981 | 0 |
| | Test | 2754 | 9 | 3 | 2 | 2753 | 2 |
| Probe | Train | 11656 | 22 | 0 | 0 | 1 | 11633 |
| | Test | 2421 | 7 | 0 | 0 | 0 | 2413 |

Table 4, shows the best results classification of the proposed method with different criteria. Results are calculated based on each class against other classes.

In order to further evaluate the above approach, the proposed system performance is compared with other methods of intrusion detection. The methods used to compare the results of their experiments on NSL-KDD data. The results of the proposed method are shown in Table 5 in comparison with the seven methods of intrusion detection.

**Table 4. Proposed IDS performance on the NSL-KDD**

| Records Type | | Accuracy | Recall | Precision | F-measure |
|---|---|---|---|---|---|
| **Normal** | Train | 99.92 | 99.89 | 99.94 | 99.92 |
| | Test | 99.82 | 99.77 | 99.87 | 99.82 |
| **DOS** | Train | 99.93 | 99.89 | 99.96 | 99.93 |
| | Test | 99.85 | 99.73 | 99.97 | 99.85 |
| **U2R** | Train | 92.27 | 99.92 | 86.66 | 92.82 |
| | Test | 98.68 | 99.83 | 97.58 | 98.69 |
| **R2L** | Train | 99.26 | 99.93 | 98.61 | 99.27 |
| | Test | 99.64 | 99.86 | 99.42 | 99.64 |
| **Probe** | Train | 99.87 | 99.93 | 99.80 | 99.87 |
| | Test | 99.77 | 99.82 | 99.71 | 99.77 |

**Table 5. Comparison proposed IDS performance with other methods (%)**

| Methods | Normal | DOS | U2R | R2L | Probe | Accuracy |
|---|---|---|---|---|---|---|
| Fuzzy+ACO [24] | - | - | - | - | - | 99.69 |
| ACO+SVM [25] | - | - | - | - | - | 98.29 |
| IDS ACO [15] | 97.41 | 99.78 | 93.51 | 99.17 | 74.65 | 98.9 |
| FARCHD [26] | 99.81 | 98.05 | 65.38 | 87.54 | 95.83 | 99.00 |
| SIPSO [21] | - | 99.80 | 97.50 | 82.50 | 99.70 | - |
| CSM [27] | - | - | - | - | - | 99.79 |
| MARS [28] | 99.71 | 99.97 | 76.00 | 98.75 | **99.85** | 92.75 |
| **My Method** | **99.87** | **99.97** | **97.52** | **99.42** | 99.71 | **99.81** |

As it is known, the proposed method is more accurate than other methods of intrusion detection and for some of the attacks, and in the remaining cases it also provides an accurate precision. In Table 5, the values of each class are based on the values calculated in the relevant research, so some fields may not be presented in the research. The results show that the proposed method works uniformly on all classes and provides the desired accuracy. The reason for this is the selection of features in a hierarchy of high-density populations.

## 5. Conclusion and Future Work

The accuracy of data mining algorithms depends on the selection of appropriate attributes and the number of records required for learning. The results show that the proposed genetic algorithm chooses appropriate features according to a hierarchical process. The precise and adaptive adjustment of the similarity parameters has led to the identification of more normal and pecliar features, which has led to the effectiveness of the proposed method. The results show that the proposed intrusion detection system has a high accuracy in detecting the intrusion of the DOS type and its underlying attacks. Also, U2R penetration is less accurate than other attacks. The reason for this is the low number of training samples used to test in the dataset. The results of the proposed method showed a precision of %99.81, which is superior to similar algorithms. Another requirement for intrusion detection systems is to find the optimal feature set for each type of attack. Because in this case, the Intrusion Detection System will be able to use only a feature set appropriate to that attack to detect any attack.

As for the future work, intention is to apply the proposed intrusion detection method using complicated classifiers to improve its performance and to combine the proposed method with other population-based algorithms. Analyzing packet payload is recently attracting lots of attention and many researchers report works carried-out in this area. It is notable that feature selection for the payload-based intrusion detection is not mature yet. Intension will be to extract and selection appropriate features from the packet payload to improve the detection rate.

## 6. REFERENCES

[1] Cisco, I., 2012. Cisco visual networking index: Forecast and methodology, 2011–2016. CISCO White paper, pp.2011-2016.

[2] Lakhina, S., Joseph, S. and Verma, B., 2010. Feature reduction using principal component analysis for effective anomaly–based intrusion detection on NSL-KDD.

[3] Bace, R. and Mell, P., 2001. NIST special publication on intrusion detection systems. BOOZ-ALLEN AND HAMILTON INC MCLEAN VA.

[4] Karabulut, E.M., Özel, S.A. and Ibrikci, T., 2012. A comparative study on the effect of feature selection on classification accuracy. Procedia Technology, 1, pp.323-327.

[5] Safavian, S.R. and Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), pp.660-674.

[6] Peterson, L.E., 2009. K-nearest neighbor. Scholarpedia, 4(2), p.1883.

[7] Liu, H., Motoda, H., Setiono, R. and Zhao, Z., 2010, May. Feature selection: An ever evolving frontier in data mining. In Feature Selection in Data Mining (pp. 4-13).

[8] John, G.H. and Langley, P., 1995, August. Estimating continuous distributions in Bayesian classifiers. In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence (pp. 338-345). Morgan Kaufmann Publishers Inc.

[9] Quinlan, J.R., 1993. C4. 5: Programs for Machine Learning Morgan Kaufmann San Mateo. CA Google Scholar.

[10] Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[11] Werbos, P.J., 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Doctoral Dissertation, Applied Mathematics, Harvard University, MA.

[12] Kratica, J., Kostić, T., Tošić, D., Dugošija, D. and Filipović, V., 2012. A genetic algorithm for the routing and carrier selection problem. Computer Science and Information Systems, (21), pp.49-62.

[13] Xu, Y., Wu, C., Zheng, K., Wang, X., Niu, X. and Lu, T., 2017. Computing Adaptive Feature Weights with PSO to Improve Android Malware Detection. Security and Communication Networks, 2017.

[14] Jovanovic, R. and Tuba, M., 2013. Ant colony optimization algorithm with pheromone correction

strategy for the minimum connected dominating set problem. Computer Science and Information Systems, 10(1), pp.133-149.

[15] Sung, A.H. and Mukkamala, S., 2003, January. Identifying important features for intrusion detection using support vector machines and neural networks. In Applications and the Internet, 2003. Proceedings. 2003 Symposium on (pp. 209-216). IEEE.

[16] Sung, A.H. and Mukkamala, S., 2004, December. The Feature Selection and Intrusion Detection Problems. In ASIAN (pp. 468-482).

[17] Vancea, F., 2014. Intrusion detection in NEAR system by anti-denoising traffic data series using discrete wavelet transform. Advances in Electrical and Computer Engineering, 14(4), pp.43-48.

[18] De la Hoz, E., de la Hoz, E., Ortiz, A., Ortega, J. and Martínez-Álvarez, A., 2014. Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. Knowledge-Based Systems, 71, pp.322-338.

[19] Sequeira, D., 2003. Intrusion prevention systems: security's silver bullet?. Business Communications Review, 33(3), pp.36-41.

[20] Anwar, S., Zain, J.M., Zolkipli, M.F., Inayat, Z., Jabir, A.N. and Odili, J.B., 2015, August. Response option for attacks detected by intrusion detection system. In Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on (pp. 195-200). IEEE.

[21] Warsi, S., Rai, Y. and Kushwaha, S., 2015. Selective Iteration based Particle Swarm Optimization (SIPSO) for Intrusion Detection System. International Journal of Computer Applications, 124(17).

[22] Gen, M. and Cheng, R., 2000. Genetic algorithms and engineering optimization (Vol. 7). John Wiley & Sons.

[23] Nsl-kdd data set for network based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSL-KDD.html, (March 2009)

[24] Varma, P.R.K., Kumari, V.V. and Kumar, S.S., 2016. Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System. Procedia Computer Science, 85, pp.503-510.

[25] Mehmod, T. and Rais, H.B.M., 2016. Ant Colony Optimization and Feature Selection for Intrusion Detection. In Advances in Machine Learning and Signal Processing (pp. 305-312). Springer International Publishing.

[26] Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S. and Herrera, F., 2015. On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. Expert Systems with Applications, 42(1), pp.193-202.

[27] Chae, H.S., Jo, B.O., Choi, S.H. and Park, T.K., 2013. Feature selection for intrusion detection using nsl-kdd. Recent Advances in Computer Science, pp.184-187.

[28] Dorigo, M. and Gambardella, L.M., 1997. Ant colonies for the travelling salesman problem. biosystems, 43(2), pp.73-81.