Recent Advances in Computer Science and Communications, 2021, *14*, 1381-1389

RESEARCH ARTICLE

Intrusion Detection System for Malicious Traffic Using Evolutionary Search Algorithm

Samar Al-Saqqa[1,*], Mustafa Al-Fayoumi[2] and Malik Qasaimeh[3]

[1]*Department of Information Technology, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan;* [2]*Department of Computer Science, King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan;* [3]*Department of Software Engineering, King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan*

Abstract: ***Introduction*:** Intrusion detection systems play a key role in system security by identifying potential attacks and giving appropriate responses. As new attacks are always emerging, intrusion detection systems must adapt to these attacks, and more work is continuously needed to develop and propose new methods and techniques that can improve efficient and effective adaptive intrusion systems. Feature selection is one of the challenging areas that need more work because of its importance and impact on the performance of intrusion detection systems. This paper applies an evolutionary search algorithm in feature subset selection for intrusion detection systems.

***Methods*:** The evolutionary search algorithm for the feature subset selection is applied and two classifiers are used, Naïve Bayes and decision tree J48, to evaluate system performance before and after features selection. NSL-KDD dataset and its subsets are used in all evaluation experiments.

***Results*:** The results show that feature selection using the evolutionary search algorithm enhances the intrusion detection system with respect to detection accuracy and detection of unknown attacks. Furthermore, time performance is achieved by reducing training time, which is reflected positively in overall system performance.

Discussion: The evolutionary search applied to select IDS algorithm features can be developed by modifying and enhancing mutation and crossover operators and applying new enhanced techniques in the selection process, which can give better results and enhance the performance of intrusion detection for rare and complicated attacks.

***Conclusion*:** The evolutionary search algorithm is applied to find the best subset of features for the intrusion detection system. In conclusion, it is a promising approach to be used as a feature selection method for intrusion detection. The results showed better performance for the intrusion detection system in terms of accuracy and detection rate.

ARTICLE HISTORY

Received: April 05, 2019
Revised: May 20, 2019
Accepted: November 28, 2019

DOI:
10.2174/2666255813999200821162547

Keywords: Intrusion detection, evolutionary search, security, NSL-KDD, feature selection, Denial of Service (DoS).

1. INTRODUCTION

With the rapid development of the internet every day, there is an urgent and never-ending need to increase the security of systems and applications to protect valuable information from attacks. Intrusion Detection Systems (IDS) are important tools that are widely used to ensure network security. They protect computers from attacks by detecting and identifying unauthorized use, modification, and destruction of data. IDS help identify unusual activities and odd behavior from the normal behavior from monitored data and recognize known attacks within a network [1-4].

Networking attacks can be classified into one of four network attack categories [5]:

1. Denial of Service (DoS): a network attack that seeks to make a machine or network resource unavailable to its intended users by making the computing or memory resource too busy or too full, which can be executed by flooding the resource with useless traffic or sending information that causes a crash. Many attacks are classified as DoS attacks such as smurf, apache2, pod, and ping of death.

*Address correspondence to this author at the Department of Information Technology, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan; Tel/Fax: +962-6-5355000;
E-mail: s.alsaqqa@ju.edu.jo

2666-2566/21 $65.00+.00

2. Probing: an attack in which the machine or network is scanned to determine weaknesses or vulnerabilities to be exploited later, such as saint, satan, nmap, portsweep, and mscan attacks.

3. Remote to User Attacks (R2L): an attack that targets one or more network resources by which attackers gain unauthorized access to network resources and try to find vulnerable points that can be used to access the network or the system in subsequent secondary attacks, such as xnsnoop, sendmail, xlock, and phf.

4. User to Root Attacks (U2R): this attack starts with legal access to a local machine and then attempts to abuse the system vulnerabilities to obtain super privileges or root access, such as a rootkit, ps, and perl attacks.

There are different approaches that are applied for IDS. The machine learning approach is followed to detect intrusion using various classification techniques using training and testing phases. The model is built then it is used to classify unknown data. Supervised, unsupervised, and hybrid machine learning approaches are applied for IDS [6]. However, feature selection is an important step that precedes building models in machine learning and which plays a role in enhancing classification and prediction. Many intrusion detection models have been developed with feature selection and classification techniques. Feature selection is an IDS preprocessing step concerned with selecting the optimal subsets of the features that represent the entire dataset. In recent years, due to large data sets containing a wide range of features, the selection of features is taken into consideration. Different selection techniques are used for intrusion detection systems because of their advantages, especially for classification results and time performance. The irrelevant and redundant features in a dataset may cause inaccurate prediction results and slow down the process of prediction or classification, so feature selection techniques play an important role not only to reduce processing time but also to select subsets of the feature that can work with a classifier to give higher classification results [7]. Since feature selection is deemed necessary to enhance the performance of IDS, it follows that there is a need for applying a variety of feature selection techniques. In this paper, we use the evolutionary search algorithm implemented by [8] to find the optimal subset of features for IDS using the most widely used dataset (NSL-KDD) with two classifiers to evaluate system performance: Naïve Bayes and decision tree J48.

The rest of the paper is organized as follows. Related work is presented in section two. Section three introduces the evolutionary search. The experiments are explained in section four.

2. RELATED WORK

A significant amount of research has been conducted in the literature to develop IDS that help to achieve better network security, most of which used various feature selection techniques [9, 10]. Chae *et al.* [11] proposed a new feature selection method using feature average of total and each class data utilizing attribute ratio, calculated by frequency and mean of features. They compared their method with three feature selection methods: correlation-based feature selection, information gain, and gain ratio using J48 decision tree classifier. The evaluation results showed that their new feature selection performed much better than other selection methods. Naïve Bayes classifier has been used to evaluate the performance of IDS using a suggested feature selection method named Feature Vitality Based Reduction Method (FVBRM), in comparison with other feature reduction methods, correlation-based feature selection, information gain and gain ratio [12]. FVBRM method was found to enhance classification accuracy more than the correlation-based method, but it needed more time. Good performance was also achieved for IDS using Artificial Neural Network (ANN) [13] and Ant Tree Miner (ATM) classifier [14].

A hybrid model for intrusion detection was developed by [15], consisting of different classifiers that help in dimensionality reduction and accuracy improvement. Another intrusion detection model was proposed by [16] with chi-square feature selection and multi-class Support Vector Machine (SVM), tested on the KDD-NSL dataset. The model gave a better detection rate and reduced false alarm rate; moreover, it reduced the training and testing time. Least Square Support Vector Machine based IDS (LSSVM-IDS), built by [17], used a feature selection method named Flexible Mutual Information Feature Selection (FMIFS), which ranked the features in terms of their relevance. They investigated the performance of their model with the KDD Cup 99, NSL-KDD, and Kyoto 2006+ datasets. The evaluation results showed that the model achieved better accuracy compared with other methods. Pham *et al.* [18] improved the performance of IDS using an ensemble model based on bagging and boosting ensemble techniques using tree-based classifiers as a base classifier. They used FVBRM and gain ratio methods for feature reduction. For feature selection, they extracted 25 features using FVBRM and 35 features using gain ratio. Their findings showed that the bagging ensemble model with J48 as the base classifier achieved the best classification accuracy when working with 35 features.

3. EVOLUTIONARY SEARCH

The evolutionary algorithm can be used to select features; it helps to explore the space of features by looking for a good subset in order to improve the classification accuracy of supervised learning techniques in classification problems where a large number of features are involved [19]. Evolutionary search is based on a genetic algorithm inspired by the natural evolution process. It is a population-based algorithm that starts with a pool of randomly generated solutions, named chromosomes, then the genetic algorithm finds the chromosomes and checks them to identify those with higher objective function values. These chromosomes are considered optimum for the reproduction phase, to hopefully reproduce a better solution (new chromosome). This process of reproduction is named the crossover and mutation process. The new chromosome will be reevaluated and the same processes will be reiterated as previously to find the best evaluation value over all the generated chromosomes; each generation of new chromosomes is called an iteration or generation. Each iteration generates a new population of solutions; these

solutions are still evaluated until it reaches the best optimum evaluation values [20, 21]. Fig. (**1**) shows the general scheme of the evolutionary algorithm [22].

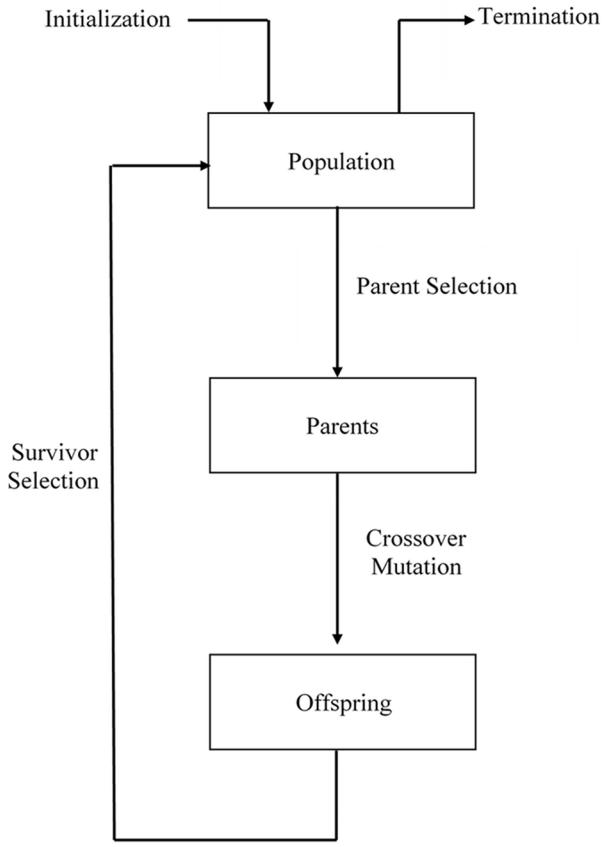

Fig. (1). A general scheme of the evolutionary algorithm.

In this research, the evolutionary search algorithm implemented by Valero [8] is applied. It performs attribute selection and implements an evolutionary algorithm that encodes individuals (solutions to the problem) with a constant length vector of bits x, where: $x(i)$ denotes that the i^{th} attribute has been selected, and $x(i) = 0$ means the opposite [23], as shown in Fig. (**2**). Evolutionary search has five main terminologies: population, number of generations, selection process, crossover process, and mutation process. The population is a set of individuals, which must be finite. The number of generations is the number of iterations. Each generation of a new individual is called an iteration or generation. Each iteration generates a new population of solutions; these solutions are still evaluated until it reaches the best optimum evaluation values. The selection process is a mechanism to select the best individuals over all of the population based on the evaluation function's resulting values. There are different techniques for selection, such as tournament selection, whereby a set of chromosomes or individuals are selected randomly from the population, then these individuals are ranked according to their relative fitness, then the fittest one wins the tournament and is selected as a parent for the next operation crossover [24]. The crossover process is the process of dividing the individual into two parts based on some criteria for two individuals, then switching the divided parts between these individuals. Single-point crossover is a classical technique used to perform crossover, in which a point in each parent is chosen randomly, and the bits of the right of that point are swapped between the two parents to produce new offspring. The mutation process is applied to the individual to give it some new values compared with the parent individuals. The flip-bit mutation operator is one of the simplest mutation operators, which simply inverts the value of the chosen bit (*i.e.* 0 goes to 1 and 1 goes to 0).

4. EXPERIMENTS/ METHODOLOGY

4.1. Dataset Description

For experiments NSL-KDD dataset is used [25]. NSL-KDD was introduced by Tavallaee [26]. It was selected in this work for different reasons. It is a revised version of the KDD Cup99 dataset, which is the most widely used. It is a standard and common benchmark for IDS evaluation [27].

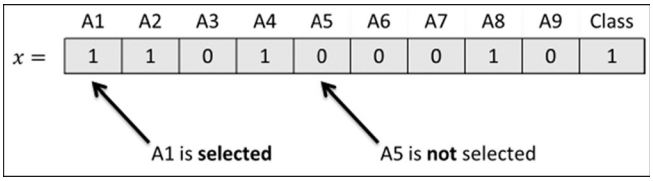

Fig. (2). Binary encoding for individuals in the evolutionary algorithm.

NSL-KDD solves some problems in KDD Cup99, including the large number of redundant records in training and testing datasets in the latter, by removing the redundant records, so the classifiers are not biased towards more frequent records [28, 29]. Moreover, the numbers of records in the training and testing are reasonable, which allows running the experiments on the complete dataset. In each record of NSL-KDD, the number of features is 41 features,), as shown in Table **1**. As in the case of KddCup99, and each record is labeled as either normal (normal traffic) or abnormal (malicious traffic. NSL-KDD dataset contains Full-KDDTrain, which contains all records of the dataset. NSL-KDD dataset also includes four sub-datasets of four types of attacks: DoS, Probe, R2L, and U2R, as shown in Table **2**. The dataset categories are changed to numerical values, and nominal fields are represented in numeric categories instead of text. The connection in the dataset is represented by many features, not all of which are needed to build IDS, so the most informative features of traffic data are selected to enhance the performance.

4.2. Experimental Settings

4.2.1. Environment

Weka 3.8.1 Package was used for experiments, and all experiments were carried out on a PC with 8 GB RAM and Intel Core i3-3217U CPU 1.8 GHz. Experiments were run on KDDTrain20 and sub-datasets per attack type, as shown in Table **2**. To evaluate the classification performance on these feature sets, we chose the Naive Bayes and decision tree J48 classifiers. 10-fold cross-validation was used to evaluate the classifiers, in which the data is divided into ten subsets, then in each of ten experiments the classifier is trained on all except

Table 1. Features description in the NSL-KDD dataset.

#	Feature Name	#	Feature Name
F1	duration	F22	serror_rate
F2	src_bytes	F23	srv_serror_rate
F3	dst_bytes	F24	rerror_rate
F4	land	F25	srv_rerror_rate
F5	wrong_fragment	F26	same_srv_rate
F6	urgent	F27	diff_srv_rate
F7	hot	F28	srv_diff_host_rate
F8	num_failed_logins	F29	dst_host_count
F9	logged in	F30	dst_host_srv_count
F10	num_compromised	F31	dst_host_same_srv_rate
F11	root_shell	F32	dst_host_diff_srv_rate
F12	su_attempted	F33	dst_host_same_src_port_rate
F13	num_root	F34	dst_host_srv_diff_host_rate
F14	num_file_creations	F35	dst_host_serror_rate
F15	num_shells	F36	dst_host_srv_serror_rate
F16	num_access_files	F37	dst_host_rerror_rate
F17	num_outbound_cmds	F38	dst_host_srv_rerror_rate
F18	is_host_login	F39	protocol_type
F19	is_guest_login	F40	service
F20	count	F41	flag
F21	srv_count		

Table 2. The number of records in each used dataset.

Dataset	No. of Features	Normal	Abnormal	Total
Full -KDDTrain	41	13449	11743	25192
DoS	41	13449	9234	22683
Probe	41	13449	2289	15738
R2L	41	13449	209	13658
U2R	41	13449	11	13460

Table 3. Confusion matrix.

Class	Predicted Abnormal	Predicted Normal
Actual Abnormal	TP	FN
Actual Normal	FP	TN

one of the subsets. The testing is held on the subset, which is not used for training, then the accuracy of the classifier model is evaluated by calculating the average of accuracies obtained in all ten cases of cross-validation.

4.2.2. Performance Measures

To measure the performance of classifiers, the following evaluation metrics are used (eqs. 1-3):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Detection\ Rate\ (DR) = \frac{TP}{TP+FN} \quad (2)$$

$$F - Measure = \frac{2*TP}{2*TP+FP+FN} \quad (3)$$

Where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. As shown in the confusion matrix illustrated in Table **3**, TP occurs in the case that an event is correctly determined as abnormal (intrusion); TN occurs when an event is correctly determined as normal; FN occurs when an event is deter-

mined as normal, but it is in actuality abnormal; and FP occurs when an event is determined as abnormal when it is actually normal.

4.2.3. Parameter Settings

In our experiments, we used the default parameters settings of evolutionary search as determined by the previous implementation in the literature [8]. For a crossover, the single-point operator is used, with a rate that equals 0.6. A total of 20 generations (iterations) is considered as a stopping criterion of the algorithm; the number of population equals 20. The initialization operator is random. For the mutation process, the bit flip is used with 1.0 rate. The selection operator is tournament selection.

5. RESULTS ANALYSIS AND DISCUSSION

The performance was evaluated before feature selection and after feature selection. WEKA CfsSubsetEval feature evaluator was used, which evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between features. Two classifiers were used, Naïve Bayes and decision tree J48, to evaluate system performance before and after features selection. Full-KDDTrain dataset and the four sub-datasets per attacks were used in experiments.

Table **4** represents the selected features after applying the evolutionary search. It can be seen that 19 features were selected for Full-KDDTrain, DoS, and R2L. For the Probe sub dataset the number of features is 17, and 12 features were selected for U2R. The number of features selected in each dataset differs due to the variety of dataset records, and the evolutionary search selecting the most relevant and important features according to records data.

In the experiments, we reported evaluation accuracy and detection rate for the full-train dataset and sub-datasets for each attack before and after the feature selection. The evaluation results using the Naïve Bayes classifier are shown in Table **5**. The results indicate that the use of the evolutionary research algorithm shows promising results in terms of accuracy, detection rate, and f-measure. Performance improvements are achieved in terms of accuracy for the Full-Train (1%), DoS (0.25%), Probe (2.65%), and R2L (0.94%) attacks, compared to the no feature selection scenario. However, there is no enhancement for U2R, which could be attributable to the number of abnormal records of this type of attack being relatively small, so the data is not well represented for this type of attack. As shown in Table **6**, the evaluation results using the J48 classifier also show improvement when using the evolutionary search algorithm to select features using a J48. Moreover, as shown from Tables **5** and **6**, J48 outperforms the Naïve Bayes classifier in terms of accuracy, detection rate, and f-measure, consistent with the J48 classifier results in the literature [15].

Table 4. The selected features in each dataset using the evolutionary search.

Training Dataset	#	Features
Full-Train	19	f2,f3,f4,f5,f9,f12,f15,f22,f23,f26,f27,f28,f29,f30,f31,f34,f39,f40,f41
Dos	19	f2,f3,f5,f8,f9,f12,f16,f20,f22,f23,f26,f27,f28,f29,f30,f34,f35,f40,f41
Probe	17	f2,f3,f7,f9,f16,f24,f27,f29,f30,f33,f34,f36,f37,f38,f39,f40,f41
R2L	19	f2,f3,f7,f8,f9,f11,f18,f19,f20,f21,f22,f28,f29,f30,f33,f34,f35,f39,f41
U2R	12	f1,f4,f6,f7,f10,f11,f14,f24,f28,f30,f33,f40

Table 5. Naïve bayes results for datasets before and after feature selection.

-	Accuracy	Detection Rate	F-Measure
Full-Train	89.59	0.9124	0.9034
Full-Train (FS)	90.59	0.9427	0.9145
DoS	96.36	0.9657	0.9692
DoS (FS)	96.61	0.9687	0.9714
Probe	88.85	0.8822	0.9311
Probe (FS)	91.51	0.9085	0.9480
R2L	94.19	0.9420	0.9696
R2L (FS)	95.13	0.9517	0.9746
U2R	90.31	0.9033	0.9488
U2R (FS)	89.36	0.8936	0.9436

Table 6. J48 results for datasets before and after feature selection.

-	Accuracy	Detection Rate	F-Measure
Full-Train	99.54	0.9954	0.9957
Full-Train (FS)	99.47	0.9955	0.9951
DoS	99.89	0.9991	0.9991
DoS (FS)	99.92	0.9994	0.9993
Probe	99.67	0.99.88	0.9981
Probe (FS)	99.68	0.9991	0.9981
R2L	99.76	0.9994	0.9988
R2L (FS)	99.76	0.9994	0.9988
U2R	99.94	1	0.9997
U2R (FS)	99.94	1	0.9997

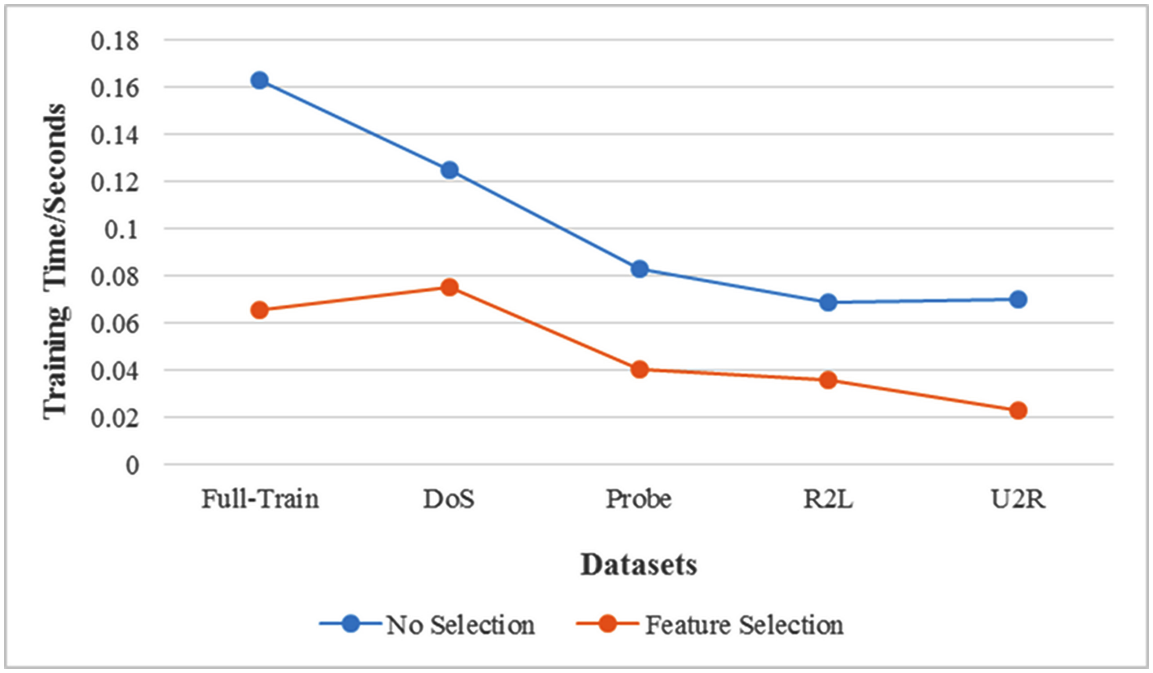

Fig. (3). Training time for naïve bayes classifier before and after feature selection. (*A higher resolution / colour version of this figure is available in the electronic copy of the article*).

Evolutionary searching also achieves computation efficiency. As shown in Figs. (**3** and **4**), the training time is reduced when using an evolutionary search in the selection of a feature compared to the training time when using all features, which is positively reflected in the overall performance of the processing.

To compare the evolutionary search with other feature selection methods, we compared the evolutionary search with Information Gain (IG) feature selection method, which is commonly used in the literature for intrusion detection [15, 30, 31]. Figs. (**5** and **6**) show the performance comparison results between evolutionary search and IG feature selection method in terms of accuracy, evaluated using Naïve Bayes and J48 classifiers. It can be seen from the figures that there is a slight difference in performance when using evolutionary as compared to the IG method, with the former performing slightly better than the latter for Full-Train, DoS, Probe, and R2L attacks. However, as shown in Fig. **5**, there is no improvement for U2R, which could be attributable to the small number of abnormal records of this type of attack, as mentioned previously.

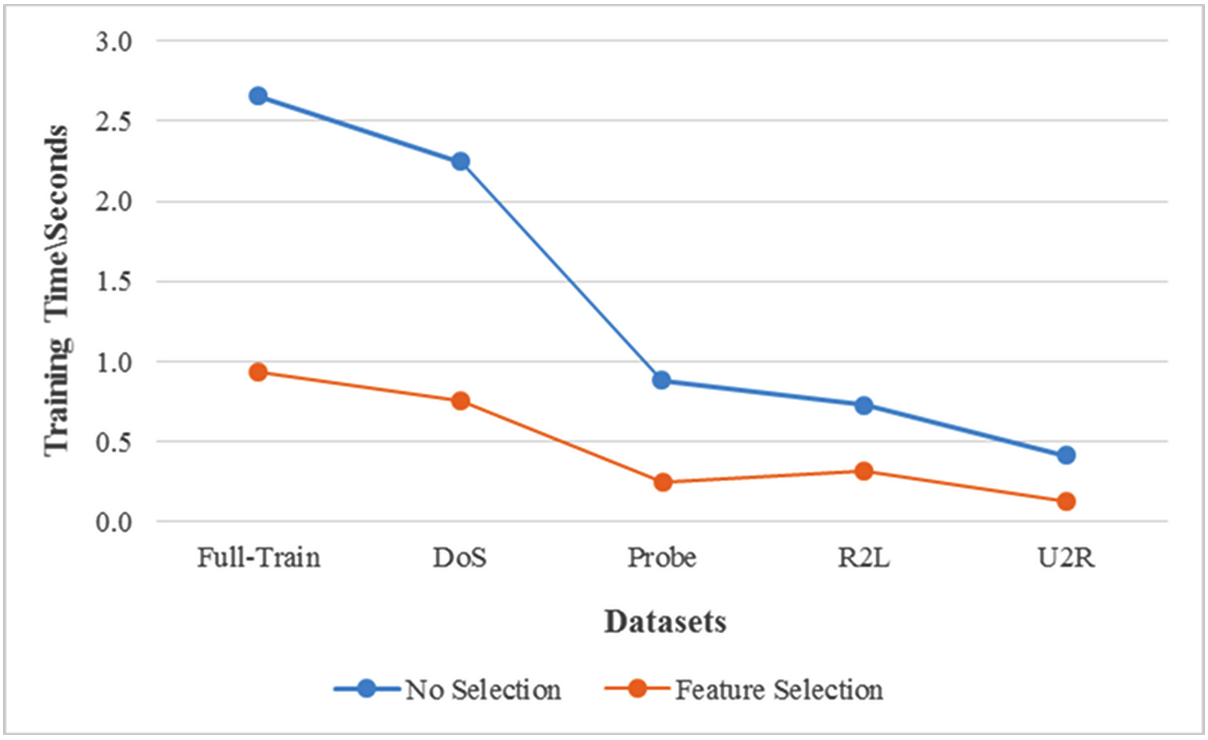

Fig. (4). Training time for the J48 classifier before and after feature selection. (*A higher resolution / colour version of this figure is available in the electronic copy of the article*).

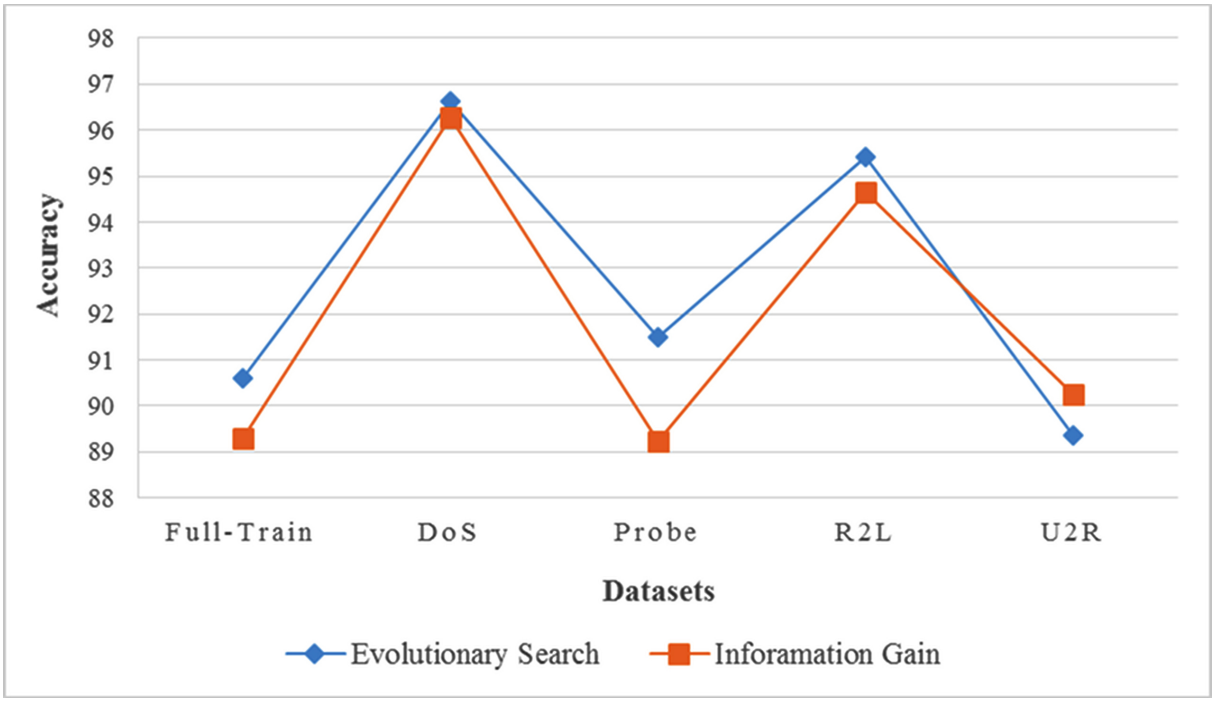

Fig. (5). Performance comparison of feature selection methods using naïve bayes classifier. (*A higher resolution / colour version of this figure is available in the electronic copy of the article*).

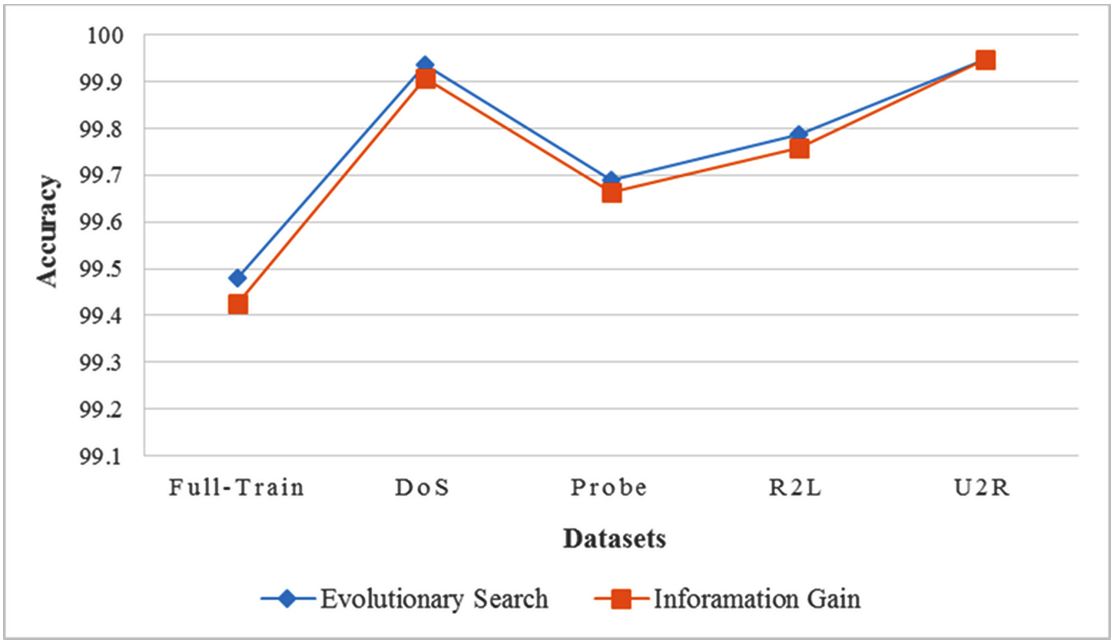

Fig. (6). Performance comparison of feature selection methods using J48 classifier. (*A higher resolution / colour version of this figure is available in the electronic copy of the article*).

CONCLUSION

IDS are very important in security infrastructure, helping detect and identify the intrusion or attack by analyzing network traffic. In this research, we applied an evolutionary search algorithm to find the best subset of features for IDS using the most widely used dataset, the NSL KDD dataset. We used two classifiers to evaluate system performance, Naïve Bayes and J48 classifiers. The results showed better performance for IDS using the evolutionary search algorithm in terms of accuracy, detection rate, and f-measure. Moreover, time performance was achieved by reducing the training time.

CURRENT & FUTURE DEVELOPMENTS

The evolutionary search applied to select IDS algorithm features can be developed by modifying and enhancing mutation and crossover operators and applying new enhanced techniques in the selection process, which can give better results and enhance the performance of intrusion detection for rare and complicated attacks.

CONSENT FOR PUBLICATION

Not Applicable.

AVAILABILITY OF DATA AND MATERIALS

All data generated or analysed during this study are included in this published article.

FUNDING

None.

CONFLICT OF INTEREST

The authors confirm that this article and its contents have no conflict of interest.

ACKNOWLEDGEMENTS

Declared None.

REFERENCES

[1] R. Kemmerer, and G. Vigna, "Intrusion detection: A brief history and overview", *Computer,* Vol. 35, pp. 27-30, 2002.

[2] M. Al-Fayoumi, Y. Ahmad, and U. Tariq, "a heterogeneous framework to detect intruder attacks in wireless sensor networks", *Int. J. Adv. Comp. Sci. Appl.,* Vol. 7, no. 12, pp. 52-58, 2017.

[3] M. Al-Fayoumi, J. Alwidian, M. Abusaif, and I. M. East, "intelligent association classification technique for phishing website detection", *Int. Arab J. Inf. Technol.,* Vol. 17, no. 4, 2020.

[4] Y. A. Yaseen, M. Qasaimeh, R. Al-Qassas, and M. Al-Fayoumi, "Email fraud attack detection using hybrid machine learning approach", *Recent Pat. Comput. Sci.,* Vol. 12, pp. 1-11, 2019. http://dx.doi.org/10.2174/2213275912666190617162707

[5] S. Paliwal, and R. Gupta, "Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm", *Int. J. Comput. Appl.,* Vol. 60, no. 19, pp. 975-8887, 2012.

[6] A. L. Buczak, and E. Guven, "a survey of data mining and machine learning methods for cyber security intrusion detection", *IEEE Comm. Surv. and Tutor.,* Vol. 18, no. 2, pp. 1153-1176, 2016. http://dx.doi.org/10.1109/COMST.2015.2494502

[7] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems", *J. Netw. Comput. Appl.,* Vol. 34, no. 4, pp. 1184-1199, 2011. http://dx.doi.org/10.1016/j.jnca.2011.01.002

[8] S. L. Valero, "An Evolutionary Algorithm (EA) to explore the space of attributes", [Online: http://neo.lcc.uma.es] [Accessed: 25-Dec-2018].

[9] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques", *J. Netw. Comput. Appl.,* Vol. 60, pp. 19-31, 2016. http://dx.doi.org/10.1016/j.jnca.2015.11.016

[10] G. Meena, and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA", In *International Conference on Computer, Communications and Electronics,* 2017, pp. 553-558. http://dx.doi.org/10.1109/COMPTELIX.2017.8004032

[11] H. Chae, B. Jo, S. Choi, and T. Park, "feature selection for intrusion detection using NSL-KDD", *Rec. Adv. Comp. Sci.,* Vol. 20132, pp. 184-187, 2013.

[12] S. Mukherjee, and N. Sharma, "intrusion detection using naive bayes classifier with feature reduction", *Procedia Technology,* Vol. 4, pp. 119-128, 2012. http://dx.doi.org/10.1016/j.protcy.2012.05.017

[13] B. Ingre, and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN", In *International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES, in Association with IEEE,* 2015, pp. 92-96. http://dx.doi.org/10.1109/SPACES.2015.7058223

[14] M. H. Aghdam, and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization", *Int. J. Netw. Secur.,* Vol. 18, no. 3, pp. 420-432, 2016.

[15] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model", *J. Comput. Sci.,* Vol. 25, pp. 152-160, 2018. http://dx.doi.org/10.1016/j.jocs.2017.03.006

[16] I. S. Thaseen, and C. A. Kumar, ""Intrusion detection model using fusion of chi-square feature selection and multi class SVM,"", *J. King Saud Uni-Comp. Inform. Sci.,* Vol. 29, no. 4, pp. 462-472, 2017. http://dx.doi.org/10.1016/j.jksuci.2015.12.004

[17] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm", *IEEE Trans. Comput.,* Vol. 65, no. 10, pp. 2986-2998, 2016. http://dx.doi.org/10.1109/TC.2016.2519914

[18] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection", In *Proceedings of the Australasian Computer Science Week Multiconference,* 2018, pp. 1-6. http://dx.doi.org/10.1145/3167918.3167951

[19] J. E. Smith, and A. E. Eiben, "What Is an Evolutionary Algorithm?", In *Introduction to Evolutionary Computing,* Berlin, Heidelberg, 2015, pp. 25-48.

[20] D. Whitley, and A. M. Sutton, "Genetic algorithms-A survey of models and methods", In *Handbook of Natural Computing,* 2012, pp. 637-671. http://dx.doi.org/10.1007/978-3-540-92910-9_21

[21] K. Jankauskas, L. G. Papageorgiou, and S. S. Farid, "Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture", *Comput. Chem. Eng.,* Vol. 121, pp. 212-223, 2019. http://dx.doi.org/10.1016/j.compchemeng.2018.09.019

[22] H. I. Calvete, C. Galé, and J. A. Iranzo, "An efficient evolutionary algorithm for the ring star problem", *Eur. J. Oper. Res.,* Vol. 231, no. 1, pp. 22-33, 2013. http://dx.doi.org/10.1016/j.ejor.2013.05.013

[23] S. L. Valero, [Online: http://neo.lcc.uma.es/25-Dec-2018]

[24] B. L. Miller, and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise", *Complex Syst.,* Vol. 9, no. 3, pp. 193-212, 1995.

[25] F. H. Botes, L. Leenen, and R. De La Harpe, "Ant colony induced decision trees for intrusion detection", In *16th European Conference on Cyber Warfare and Security,* 2017, pp. 53-62.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In *IEEE Symposium on Computational Intelligence for Security and Defense Applications,* 2009, pp. 1-6. http://dx.doi.org/10.1109/CISDA.2009.5356528

[27] S. J. Stolfo, F. Wei, W. Lee, A. Prodromidis, and P. K. Chan, "*KDD cup knowledge discovery and data mining competition*". *1999.,* 2006 [Online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html].

[28] P. Aggarwal, and S. K. Sharma, "Analysis of kdd dataset attributes-class wise for intrusion detection", *Procedia Comput. Sci.,* Vol. 57, pp. 842-851, 2015. http://dx.doi.org/10.1016/j.procs.2015.07.490

[29] A. R. A. Yusof, N. I. Udzir, A. Selamat, H. Hamdan, and M. T. Abdullah, "Adaptive feature selection for Denial of Services (DoS) attack", In *IEEE Conference on Applications, Information and Network Security,* 2017, pp. 1-4.

[30] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, "Feature selection using information gain for improved structural-based alert correlation", *PLoS One,* Vol. 11, no. 11, 2016. http://dx.doi.org/10.1371/journal.pone.0166017 PMID: 27893821

[31] I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier", *Expert Syst. Appl.,* Vol. 88, pp. 249-257, 2017. http://dx.doi.org/10.1016/j.eswa.2017.07.005