A Hybrid IDS Using GA-Based Feature Selection Method and Random Forest

Zhiqiang Liu and Yucheng Shi

***Abstract*—In recent years, the rapid development of internet technology brings many severe network security problems linked to malicious intrusions. Intrusion Detection System is considered to be one of the significant techniques to safeguard the network from both external and internal attacks. However, with the fast expansion of the IoT network, cyberattacks are also changing quickly, and many unknown types are showing up in the contemporary network environment. Consequently, the efficiency of traditional signature-based and anomaly-based Intrusion Detection System is insufficient. We propose a novel Intrusion Detection System, which uses an evolutionary technique based feature selection approach and a Random Forest-based classifier. The evolution-based feature selector uses an innovative Fitness Function to select the important features and reduces dimensions of the data, which raise the Ture Positive Rate and reduce the False Positive Rate at the same time. With exceptional high accuracy in multi-classification tasks and outstanding capabilities of handling noise in massive data scenarios, the Random Forest technique is widely used in anomaly detection. This research proposes a framework that can select more steady features and improve the classification results as compared with other technologies. The proposed framework is tested and experimented on UNSW-NB15 datasets and NSL-KDD datasets. Various statistical results and detailed comparison to other methods are presented within this article.**

***Index Terms*—Genetic algorithm, network security, NSL-KDD, random forest decision tree, UNSW-NB15.**

I. Introduction

As a result of the world's third industrial revolution, computers and networking technologies exploded in our daily life. With the convenient they brought, these technologies also left us with concern and risks. Virus, Trojan, and Worms can easily inject into our system. Sensitive information can be leaked or hijacked by cyberattacks. And all these threats still escalate with the development of information technology. The traditional defense system can identify some attacks, but as they varied a bit, they can hardly be recognized. Thus, the whole industry is seeking new mechanisms that can accurately capture and block those threats and guarantee our system in a working and safe environment.

Defense mechanisms can be categorized into Intrusion Prevention Systems (IPS) and Intrusion Detection System (IDS). Intrusion Detection System, as an entrance, usually works at the frontier of the network. According to different techniques, misuse detection and anomaly detection are two main categories in intrusion detection. Misuse detection uses known attack methods that have been defined in advance. The system determines the existence of these attacks to achieve the detection process, which is also called feature detection [1]. Misuse detection is built on the existing feature library or feature database. It can detect the intrusion patterns recorded in the signature database with high accuracy. However, misuse detection fails to detect the zero-day attack. In other words, while there are attacks which not exist in the signature database, this detection system can hardly capture them. When an alarm is raised, which means a recorded signature has been detected, though note that the set of signatures could contain ambiguous outlines that can be caused by an attacker as well as a legitimate user. Anomaly detection does not rely on the signature database. It analyses the network traffic by calculating the deviation from the user's behavior to the normal profile. Anomaly detection can address the reliance issue on the signature database, but this method may detect the normal network behavior as an intrusion, and the false alarm rate is relativity high.

Typically, an intrusion detection system consists of two components that join together. The first component selects only the necessary features, and the second component is for classification and makes efficient decisions. To achieve the best performance, these components must work along with each other to perform a low time consuming and high accuracy result.

Data pre-processing goes for a vital step at the beginning of the whole detect process. Selecting the significant information and features from the dataset can reduce the dimensions of the raw dataset, which always leads to better performance. The Genetic Algorithm (GA) is inspired by a natural evolutionary theory put forward by Darwin. Genetic Algorithm is a commonly used method for finding an optimized and high-quality solution to search problem. Core operators in GA are inspired by biological processes such as crossover, mutation, and selection [2]. Fitness function is considered to be the most important part of Genetic algorithms. The Fitness Function evaluates every offspring chromosomes, and then only the highest scored one can survive to the next evolutionary round. The disadvantage of the previous proposed Fitness Function in GA-based feature selection model is simply using the Accuracy and selected feature numbers as parameters. Ignoring the high False Positive Rate (FPR) of the intrusion detection commonly results in a low True Positive Rate (TPR).

In recent decades, machine learning has been increasingly used as another vital component of the modern Intrusion Detection System. Generally, machine learning can be divided into supervised and unsupervised learning. Supervised learning is a powerful tool in analyzing the

Manuscript received October 18, 2019; revised April 11, 2021.

The authors are with the School of Software and Microelectronic, Northwestern Polytechnical University, Shannxi, China (e-mail: zqliu_edu@163.com, shiyucheng_edu@foxmail.com).

doi: 10.18178/ijmlc.2022.12.2.1077

high-dimensional data and figuring out the hidden pattern behind these statistics. Supervised learning also has a strong capability of classifying high-dimensional data into specific classes. Thus, this technology can be used to recognize malicious behaviors in network traffic. Because of the massive, high-dimensional and strong non-linear traffic data, some classical machine learning methods, for instance, Probability-based Bayesian, Decision Tree, and Support Vector Machine (SVM), are proven to be less effective in the classification task. The results have low accuracy but a high False Positive Rate, and the "dimensional explosion" problem is prone to occur.

Random Forest (RF) is a supervised learning algorithm. After the training process with given features and classification results, an RF model can be obtained to classify new datasets. Among all kinds of supervised learning algorithms, Random Forest has certain advantages in accuracy and training speed. Also, good noise processing ability and high stability make the Random Forest a popular choice in the Intrusion Detection System. There are a number of factors to evaluate the performance of Random Forest, including accuracy, recall rate, running time, etc. We propose a model that combines the Genetic algorithm with the Random Forest algorithm to reach the best results. Besides, a newly designed Fitness Function, which adds FPR as a penalty parameter, aims to cut the false alarm rate (FAR) and increase the TPR concurrently. Furthermore, F1-score is also used for balancing the weight of the precession rate and the recall rate. We mainly optimize the accuracy and time complexity of Random Forest through parameter adjustment and data dimensionality reduction. A stable number of selected features and low decision time are also treated as an important performance indicator.

Rest of the paper is organized as follows: In Section II, related works are reviewed, in Section III, the details of the proposed Intrusion Detection System is given. Section IV discusses the experimental result in UNSW-NB15 [2] dataset compared to the NSL-KDD [3] dataset. Conclusions and some possible future enhancements on this work are presented in Section V.

II. Related Work

There is a great deal of previous researches in the literature that discussed the Intrusion Detection System. Denning D.E [4] initially proposed the abstract model of the intrusion detection system in 1987. This paper firstly uses intrusion detection as a security defense technique of the computer system. The model is independent of any specific operating system, application environment, system vulnerability and intrusion type. It is a framework that can be an excellent example of designing intrusion detection application systems. Although the audit criteria in the proposed model can be triggered by other unknown factors that are not anomalies behaviors. And the fact that whether the model can detect the most intrusion before severe damage is done still needs to be proved. Wu *et al.* [5] work intensively in database intrusion, especially in anomaly detection based on data mining, the author also uses association rules to a forward implementation based on Trie tree. Aumreesh *et al.* [6] give a review that emphasizes various types of Intrusion Detection System, such as misuse based, anomaly-based, host-based, network-based and hybrid-based. It mainly focuses on anomaly-based and behavior based along with agent-based technology in real network traffic. S. Northcutt *et al.* [7] compare the pros and cons of the anomaly detection approach and misuse detection approach respectively. The author points out that the drawback of the anomaly detection approach is that when the Intrusion Detection System experiences a new behavior for the first time, it raises the alarm, which may be a false positive. Also, the false negative rate and False Positive Rate and anomaly detection are relatively much higher than misuse detection.

L. Haripriya and M.A. Jabbar [8] give a review of using Machine Learning (ML) technologies in the Intrusion Detection System. They also discuss applications into a system with ML, and the detailed comparison of various approaches for the Intrusion Detection System using ML is given. This paper indicated that It is relatively hard to train the ML models while a certain amount of traffic data is insufficient or not available. A useful intrusion detection system model uses Artificial Neural Network (ANN) is presented by Basant Subba *et al*. [9]. One limitation in their approach is that the model they proposed requires large training time. However, the overall detection performance of the neural network will not be degraded by the failure of adding new agents to the previous one. Pan-Shi Tang *et al.* [10] describe Filter and wrapper, which is the most common feature selection algorithm in their work. A combination of two algorithms is also compared with the Genetic Algorithm based selection method, then comes out a conclusion that GA has much higher efficiency than Filter and Wrapper algorithm in selecting features. S. Aksoy *et al*. [11] and B. Kavitha *et al.* [12] describe an essential method of selecting the required subset of features by using the Genetic Algorithm. They believe feature selection can discard redundant items, as well as have a considerable effect on building efficient classification system in further steps. Ketan Sanjay Desale and Roshani Ade [13] propose an innovative feature selection technique that using a method based on mathematical intersection principle and genetic algorithm. Besides, a range type of feature selection techniques, for instance, IG, CAE, and CFS, are tested. Their outcomes of the other two regularly used classifiers, J48, and Naive Bayes (NB) are compared. These articles give a good example of using the Genetic Algorithm as a feature selector.

Yi Yi Aung *et al.* [14] develop an IDS for identifying network behavior by using K-Means and RF. Decreasing the CPU and memory consumption is also one of their focuses. Moreover, the hybrid model shows a superior to the system only using a single Random Forest algorithm, specifically in detection correctness and classification accuracy. In this work, 10% of the KDDCUP99 [15] dataset is used to testify the model accuracy. Yaping Chang *et al.* [16] apply Random Forest to select important features and SVM to improve the classification result. And only 14 features (in total 41 features) are selected to reach a higher attack detection rate, also using the KDDCUP99[16] dataset. A data mining based intrusion detection framework combining misuse and anomaly detection, which also applies the RF, is proposed by Mohammad Zulkernine and Jiong Zhang [17]. They utilize sampling techniques and optimal arguments in their

framework to increase the detection correctness of minority intrusions. Although, the first shortcoming of their work is that the hybrid system can be undermined if intrusions are much more than normal data in a dataset. Second, some high degree similar intrusions cannot be correctly detected as outliers by the system. Third, their tests and experiments still work on the KDDCUP99 [15] dataset, which is outdated and cannot truly represent the modern comprehensive network traffic. M. Zhao *et al.* [18] use GA to optimize parameters of Support Vector Machine simultaneously. The model selects optimized features and best SVM parameters by concatenating them into one chromosome. However, Fitness Function in their evolutionary process only allows the accuracy and the True Positive Rate to assess every chromosome. Extra computing time is also required in every evolutionary step.

III. Proposed GA-RF IDS Framework

The overall system architecture of proposed GA-RF IDS framework is shown in Fig. 1.

We use the Genetic Algorithm based feature selection method to select useful features. In the Genetic Algorithm, different combinations of features are called chromosomes and every chromosome will be evaluated by the Fitness Function. According to the fitness value, only the highest scored chromosome can survive to the next evolution round. The new chromosome will replace the old one in the total chromosome pool, which is called the initial population. When evolutionary loop stops, relatively characteristic features are selected out as an output of the Genetic Algorithm. On top of that, Random forest s used for further feature selection and results classification. Random Forest is considered to be a powerful tool when dealing with complex data, whether in binary classification or multi-class classification.

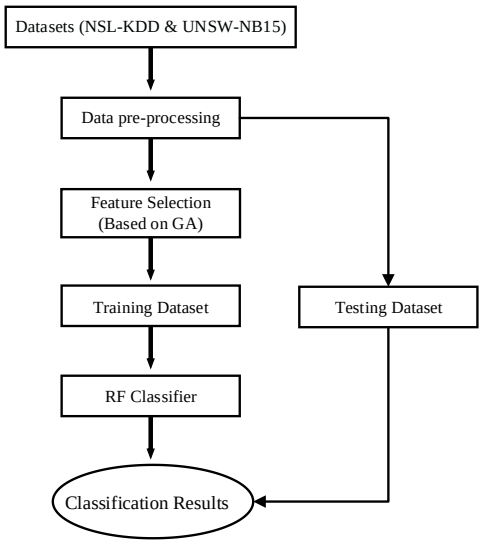


Fig. 1. The architecture of the proposed GA-RF IDS.

A. Brief Comparison of NSL-KDD and UNSW-NB15

According to UNSW-NB15 [2] dataset, the NSL-KDD [3] dataset is regarded as an upgraded version of the KDDCUP99 [16] dataset. NSL-KDD[3] dataset removes the unnecessary items in KDDCUP99 [16] and addresses the unbalancing issue among all records in both training dataset and testing dataset, which makes the detection results more reliable. NSL-KDD [3] training dataset covers 22 types of cyberattacks divided into four classes: Denial of Service (DOS), Probing Attack (PROBE), User to Root (U2R), and Remote to User (R2L). Table I presents the detail categories of all attack types. Table I. also gives a brief description of different classes. Fig. 2 shows the distribution of normal traffic and 4 types of abnormal traffic. It clearly illustrates that the percentage of records in the dataset is inversely proportional to the number of records in each difficulty level.

TABLE I: Categories of Various Attack in KDD

Class	Description	Attack Subclass
DoS	Restrict or deny a legitimate user request to a system	'smurf', 'back', 'Neptune', 'pod', 'teardrop', 'land'
PROBE	Identify and gather vulnerabilities exposed in a system or a network device	'Ipsweep', 'nmap', 'portsweep', 'satan'
U2R	Pretend to be a legitimate user or gain unauthorized Root access to a system	'loadmodule', 'buffer_overflow', 'rootkit', 'perl'
R2L	Gain unofficial local access from a remote machine	'warezmaster', 'guess_password', 'imap', 'phf', 'spy', 'multihop', ' ftp_write , 'wareaclient'

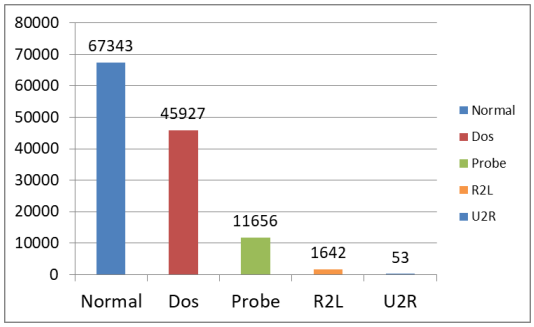


Fig. 2. Distribution chart of category in KDD training dataset.

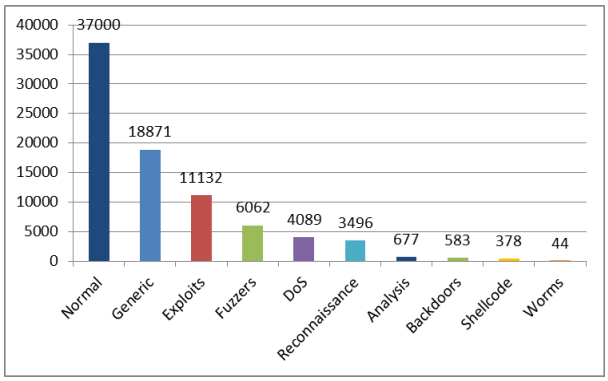


Fig. 3. Attack distribution in UNSW-NB15 (training dataset).

However, according to UNSW-NB15 [2], NSL-KDD [3] dataset does not represent the current low footprint attack scenarios. The UNSW-NB15[2] dataset is created to serve an all-inclusive environment of the contemporary network traffic, by establishing the synthetic network using the IXIA tool, which can generate real current regular traffic and synthetically abnormal traffic. UNSW-NB15 [2] dataset has 49 features though NSL-KDD [3] dataset only has 41 features. Moreover, the extra features can be regarded as key features and show benefits in previous work. All of the records in UNSW-NB15 [2] are categorized into ten groups, which are

Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Table II gives detailed description of all attack types in UNSW-NB15 [2]. And Fig. 3 illustrates the distribution of the training dataset.

In this paper, the experiment will be executed on each dataset, and results are presented in Section IV.

TABLE II: DETAILED INFORMATION OF ATTACK TYPES IN UNSW-NB15

Class	Description	Attack Subclass
Fuzzers	Attempts to suspend a program or network by providing randomly generated data.	24246
Analysis	Contains different attacks of port scan, spam and html files penetrations.	2677
Backdoors	A technique that bypasses system security to access a computer or its data.	2329
Dos	Restrict or deny a legitimate user request to a system	16353
Exploits	An attacker knows about a security problem in an operating system or software and uses the vulnerability to exploit that knowledge.	44525
Genetic	One technique applies to all block ciphers with a given block and key size, regardless of the structure of the block ciphers.	215481
Reconnaissance	Includes all strikes that can simulate attacks that collect information.	13987
Shellcode	A small piece of code that exploits software weaknesses as payloads.	1511
Worms	Using security failures to replicates itself in order to infect other computers.	174

B. Data Pre-processing

Pre-Processing transforms the data in a uniform format. It also used to remove the useless data, which is not required for the proposed method and to complete the missing data.

1) 1-N encoding

To evaluate a model, UNSW-NB15 [2] and NSL-KDD [3] are used as benchmark dataset. All the relevant experiments are performed using the mentioned datasets above. Moreover, using only the material and crucial features to classify the data source is essential. For better results of feature selection, NSL-KDD [3] dataset and UNSW-NB15 [2] dataset cannot be used to train directly as the existence of non-numeric features in datasets. To overcome this problem, non-numeric features are converted into numeric features by using 1-n numeric coding. In this paper, all the non-numeric features like protocol, service, and flag have been converted into numeric features. For example, the protocol type feature in NSL-KDD [3] consists of 3 nominal values which are tcp, udp, and icmp, the string value 'tcp' is replaced by 1, 'udp' by 2 and 'icmp' by 3 et.

2) Normalization

Features in both datasets like "src-bytes", "dst-byts", "duration" etc. ranges from 0 to 500000, which make the dataset unbalanced and unfit to be processed. Unrivaled records in the dataset will mislead the classifier and result in an inexact outcome. Therefore, these values or features should be normalized by using the following Max-Min (1) function:

$$\frac{x_i - Min}{Max - Min} \tag{1}$$

The above equation represents the normalization process, where the Minimum value and Maximum value from all available data x_i represents each data point.

3) SMOTE algorithm

Because of the minority of some specific cyberattack types, such as R2L and U2R in NSL-KDD [3] dataset, Worms and Shellcode in UNSW-NB15 [2] dataset, standard classifier always detect those cyberattacks with very low accuracy. Synthetic Minority Oversampling Technique (SMOTE) is used to overcome this problem. SMOTE is considered to be an improved approach based on the Random Oversampling algorithm. Primarily the SMOTE Algorithm utilizes the K-Nearest Neighbor (KNN) to generate the new samples, from a relatively small number of samples, mapped to the original dataset. The algorithm step is shown below:

1) For *every* sample x in the minority class S_{min}, calculate the Euclidean Distance for each of the rest in S_{min} to obtain its K-Nearest Neighbor (KNN).
2) For *each* minority sample x, randomly select several samples from its k-Nearest Neighbors, assuming that the selected neighbor is x_n.
3) For *each* x_n , construct a new sample using the following formula:

$$x_{new} = x + rand(0,1) \times |x - x_n| \tag{2}$$

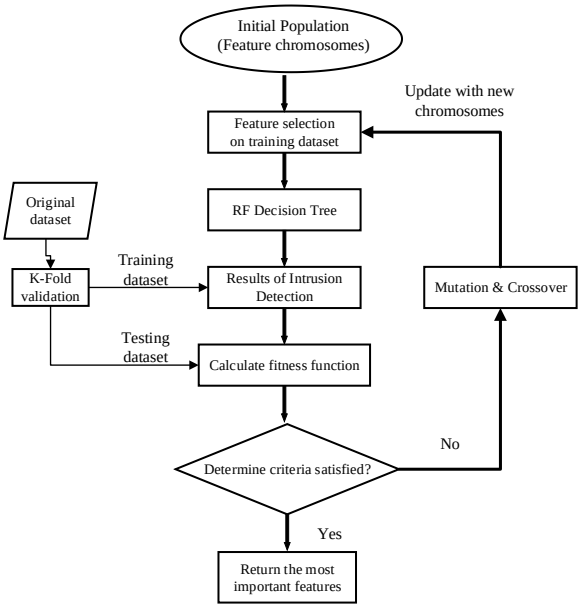


Fig. 4. Workflow chart of feature selection.

C. Genetic Algorithm Based Feature Selection Method

In the proposed method, we use the Genetic Algorithm [19] as the base of the feature selection method. Fig. 4 illustrates the workflow of our proposed feature selection process. Initial population consists of the feature chromosomes. Features in NSL-KDD [3] and UNSW-NB15 [2] dataset are coded into binary formation such as 110110111…00101101. The chromosomes are generated randomly. On the other hand, to include as many categories of attack as possible in both datasets, the number of the initial population is restricted in 100 to 150. According to the previous researches, the larger the initial population it is, the more complex the algorithm is, and more computing time is needed. On the contrary, if the initial population is too small, the optimal performance of the

algorithm will be reduced, and it is easy to fall into the local optimal solution. Both original datasets are separated into training and testing datasets by using the K-Fold validation method during the training process. Mutation rate and crossover rate are kept constant in experiments. Based on the classification results by the RF, Fitness Function evaluates every chromosome at the end of the iteration. When any of the following conditions are satisfied, the feature extraction algorithm terminates:

1) When the maximum number of preset iterations is reached, the search is complete.
2) The maximum fitness value does not change for 10 successive generations.

D. The Fitness Function

Fitness Function (8) is considered to be the most vital and fundamental part of the genetic algorithm to evaluate a chromosome to survive. At the end of every evolutionary step, the highest scored chromosome evaluated by Fitness Function will replace the lower scored one. A proper Fitness Function should preserve chromosomes with high fitting values and speed up the iterative process of the genetic algorithm. Moreover, in the Intrusion detection system scenario, notably, not only the accuracy and the True Positive Rate should be considered, but also the False Positive Rate should be included in the Fitness Function. Previously, researchers select subsets with higher classification accuracy and fewer features. However, they did not take false detection in, so those feature subsets would result in higher false alarm rates, and the performance of the Intrusion Detection System would degrade.

E. Random Forest Decision Tree

Random forest is considered to be an integrated learning method based on decision trees. The Random Forest was proposed by Leo Breiman in 2001 to combine the bagged integrated learning theory [20] with the random subspace method [21]. RF is a well-known classifier for supervised learning. In the RF decision tree, each node is classified on the bases of optimal feature selection. This process continues until we reach the termination criteria. Each node categorized as the relatively same kind of data. The number of votes determines the classification result. The most voted leaf node is considered to be the category of the sample. The voting process determined by path moving from root node to leaf node. The resistance of RF to noise and outliers not only solve many performance issues but also give us good stability. The Non-Parametric nature of RF makes it a better choice for the classification of high-dimensional data.

F. Proposed Fitness Function

We propose an innovative Fitness Function (8) which uses three parameters named Accuracy, F1-score and False Positive Rate (FPR) to evaluate each chromosome feature.

1) True Positive (TP): Classify the samples that originally belong to positive categories into positive categories.
2) True Negative (TN): Classify the samples that originally belong to negative categories into negative categories.
3) False Positive (FP): Incorrectly classify the samples that originally belong to negative categories into positive categories.
4) False Negative (FN): Incorrectly classify the samples that originally belong to positive categories into negative categories.

Accuracy (3) is the percentage of data that is correctly predicted. Accuracy is calculated as below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$\mathrm{F}_1 - score$ Eq. (4) is calculated as follows:

$$\mathrm{F}_1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

In the data that predicted to be positive, the ratio of actually positive data is called precision Eq. (5). In the actually positive data, the ratio of data that predicted to be positive is called recalls Eq. (6). The formula of precision and recall is shown below:

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$recall = \frac{TP}{TP + FN} \tag{6}$$

FPR Eq. (7) is the rate of the false positive detection calculated by:

$$FP\ rate = \frac{FP}{FP + TN} \tag{7}$$

The formula of the Fitness Function Eq. (8) is as below:

$$\mathrm{Fitness}(\mathrm{c}) = w_a \times RF_{Accuracy} + (w_b \times \mathrm{F}_1 - score) - w_c \times \mathrm{FPR} \tag{8}$$

In the proposed Fitness Function Eq. (8), w_a weights for accuracy of Random Forest Decision Tree, w_b weights for $F_1 - score$ and w_c weights for False Positive Rate. The $F_1 - score$ is a measure of test accuracy. It is the harmonic mean of precision and recall, which takes both precision and recall of the classification model into account to compute. $F_1 - score$ reaches its best value at 1 (perfect precision and recall) and worst at 0.

We assume the high False Positive Rate leads to a False alarm, which could make the Intrusion Detection System judge normal network traffic to a malicious one. We propose to increase the TPR and decrease the FPR simultaneously. Therefore, we treat the False Positive Rate as a penalty parameter in our Fitness Function, which means a high False Positive Rate makes a lower value of the whole Fitness Function. Every chromosome is evaluated by the proposed Fitness Function at the end of every loop shown in Fig. 4 and only high scored chromosome can survive to next evolutionary round.

IV. Experiments and Results

The testbed of our proposed method is a Windows platform based computer of hardware configuration having Intel Core i7-8th generation in 2.3GHz and 8 GB RAM. DEAP framework (version 1.28) was used to perform the Genetic Algorithm under Python. Detail parameters of the

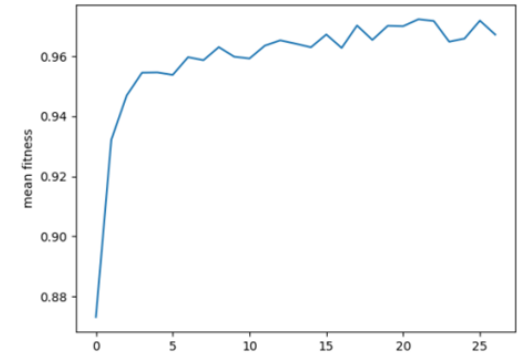
Genetic Algorithm and Fitness Function are shown in Table III.

TABLE III: DETAILED PARAMETERS IN GA AND FITNESS FUNCTION

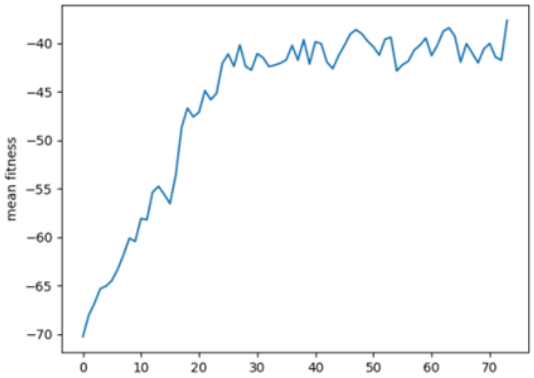
Evolution parameters	
Parameters Name	Number
Initial population	150
Mutation rate	0.01
Crossover rate	0.75
Selection type	Roulette wheel selection
Crossover type	Two-point crossover
Fitness Function parameters	
w_a	0.6
w_b	0.4
w_c	100

According to the Fitness Function (8), fitness score can be influenced by different values of parameters. After many experiments, we found mean fitness value reached its peak when w_a =0.6 and w_b =0.4. We defined the DEAP framework to be a problem of Maximization and set w_c=100 to amplify the weight of FPR to achieve the best result. The overall mean fitness value in NSL-KDD [3] dataset (a) and UNSW-NB15 [2] dataset (b) are shown in Fig. 5. The X-axis represents the N×10th generation of the loop, and Y-axis represents the mean fitness values of each generation. As shown in Fig. 5, with the process of chromosome selection, the function graph shows an upward trend and then gradually flattens out, which means right scored chromosomes are preserved in population and important selected features are slowly becoming steady.

$F1-score$, Accuracy, Recall, Precision, and FPR for both NSL-KDD [3] Train dataset and UNSW-NB15 [2] Train dataset in binary-classification are shown in Fig. 6. And the ROC Curve for both datasets is shown in Fig. 7.



(a)



(b)

Fig. 5. (a) Mean fitness in NSL-KDD. (b) Mean fitness in UNSW-NB15.

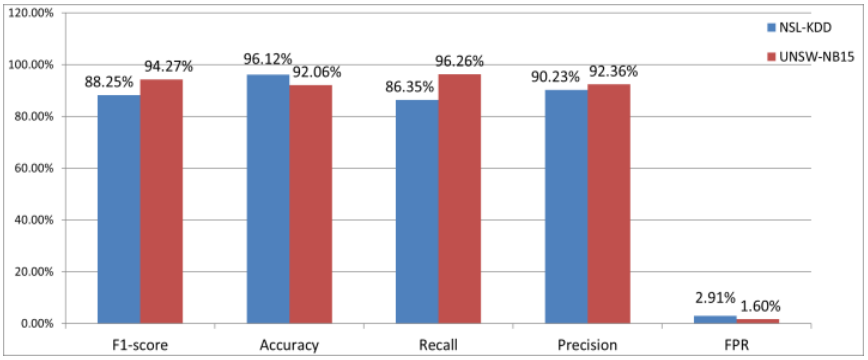
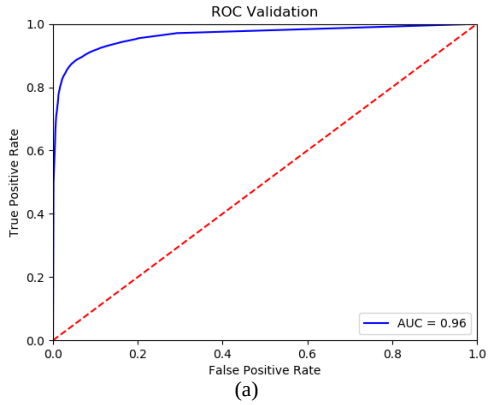
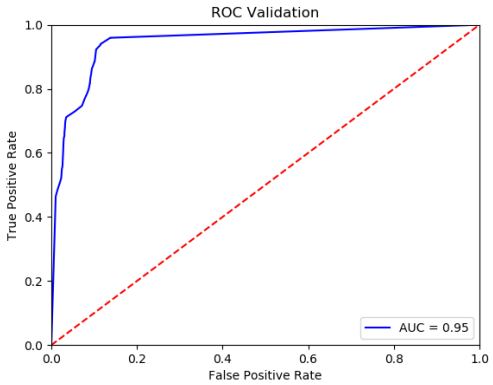


Fig. 6. Evaluate index for NSL-KDD and UNSW-NB15.

Combining the feature selection results from the Genetic Algorithm and the Random Forest, Table IV collects important features for binary-classification and multi-class classification in NSL-KDD [3] dataset and the UNSW-NB15 [2] dataset.



(a)



(b)

Fig. 7. (a) ROC Curve for NSL-KDD. (b) ROC Curve for UNSW-NB15.

Accuracy and AUC can reflect the classification ability of the classifier. Due to the imbalance problem in NSL-KDD [3] testing dataset and UNSW-NB15 [2] testing dataset, AUC

number can show the classification ability of the framework more objectively. Performance in NSL-KDD [3] dataset and UNSW-NB15 [2] dataset is shown in Table V.

TABLE IV: SELECTED FEATURES FOR NSL-KDD AND UNSW-NB15

Result with NSL-KDD dataset		
Class	Numbers	Selected Features
Normal	12	1,2,3,4,5,6,7,10,11,12,30,36
DOS	14	29,30,23,5,4,38,6,35,25,24, 36,26,39,2
PROBE	15	36,5,35,33,12,2,40,37,6,3, 32,27,41,30,26
R2L	11	23,3,5,33,12,24,10,36,32,37,6
U2R	12	1,24,33,32,36,23,6,10,14,17,5,13
Result with UNSW-NB15 dataset		
Normal	9	27,3,41,35,36,10,31,2,18
Reconnaissance	14	41,36,27,31,8,7,28,33,10, 34,40,6,15,13
Exploits	8	41,31,27,28,7,2,13,14
Fuzzers	11	10,3,4,41,36,31,28,29,45,46,47
Worms	9	41,36,7,3,39,27,29,31,10
Generic	9	35,7,3,2,27,9,11,33,46
Shellcode	7	36,44,33,34,8,10,45
Dos	12	2,27,41,36,31,7,12,3,10,43,45,47
Analysis	7	27,2,35,7,12,28,36
Backdoor	10	35,27,2,33,14,9,17,25,23,42

TABLE V: PERFORMANCE IN NSL-KDD AND UNSW-NB15

Result with NSK-KDD Testing dataset			
class	Accuracy (%)	FPR (%)	AUC
Normal	96.12	2.91	0.96
Dos	97.31	1.49	0.98
PROBE	94.58	1.39	0.96
R2L	90.79	0.07	0.92
U2R	88.21	0.11	0.85
Result with UNSW-NB15 Testing dataset			
Normal	92.06	1.60	0.95
Reconnaissance	91.24	0.60	0.94
Exploits	94.69	1.62	0.95
Fuzzers	86.04	2.10	0.91
Worms	98.81	1.14	0.98
Generic	99.25	0.39	0.99
Shellcode	95.43	2.49	0.97
Dos	94.03	2.06	0.90
Analysis	90.35	0.82	0.87
Backdoor	86.92	2.81	0.82

Compared with other technologies, our proposed GA-RF Intrusion Detection System shows more effectiveness testing on NSL-KDD [3] dataset and the UNSW-NB15 [2] dataset, which can highly represent the current network traffic state. The performance comparison is shown in Table VI.

TABLE VI: PERFORMANCE COMPARED WITH OTHER METHODS

Method	Accuracy (%)	FPR (%)	DATASET
ANN [9]	98.86(three layer)	-	NSL-KDD
GA-based J48[13]	91.86	-	NSL-KDD
GA-based NB [13]	89.5	-	NSL-KDD
K-mean RF [14]	99.8	-	10% of KDD'99
RS-GA-SVM [16]	88.2	2	KDD'99
RF-based IDS [17]	94.7	2	KDD'99
GA-RF (Proposed)	96.12	2.91	NSL-KDD
GA-RF (Proposed)	92.06	1.60	UNSW-NB15

V. CONCLUSIONS

In this paper, we propose a novel Genetic Algorithm based feature selection Intrusion Detection System which uses the Random Forest classifier. This evolutionary algorithm is used to select optimal features for the intrusion dataset. A new Fitness Function for the Genetic Algorithm is designed to achieve high TPR and low FPR at the same time. We also propose an optimized Random Forest classifier, which combining the Genetic Algorithm based feature selection method, and showing higher accuracy and AUC in both binary-class classification and multi-class classification. FPR is also lower than other techniques. Two benchmark datasets, NSL-KDD [3] dataset and UNSW-NB15 [2] dataset, are run in experiments, though the UNSW-NB15 [2] dataset is considered as a more effective representation of modern network traffic. SMOTE algorithm is used for both NSL-KDD [3] training dataset and UNSW-NB15 [2] training dataset, which can remarkably improve the detection correctness of minority attacks. The main advantage of our proposed framework is that it improves the detection accuracy of the classic Random Forest by selecting essential features and reducing training time.

Future work will be focused on GPU computing to shorten training time. Some deep learning algorithms will also be considered to improve detection accuracy further.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

ZhiQiang Liu conducted the research; YuCheng Shi analyzed data and wrote the paper.

REFERENCES

[1] W. K. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. the 1999 IEEE Symposium on Security and Privacy*, 1999, pp. 120-132.

[2] N. Moustafa, "UNSW-NBI5: A comprehensive data set for network intrusion detection systems (UNSW-NBI5 network data set)," in *Proc. Military Communications and Information Systems Conference (MiICIS)*, 2015.

[3] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, pp. 1848-1853, 2013.

[4] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222-232, 1987.

[5] W. Gongxing and H. Yimin, "Design of a new intrusion detection system based on database," in *Proc. 2009 International Conference on Signal Processing Systems*, 2009, pp. 814-817.

[6] A. K. Saxena, S. Sinha, and P. Shukla, "General study of intrusion detection system and survey of agent based intrusion detection system," in *Proc. 2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 421-471.

[7] S. Northcutt and J. Novak, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3, pp. 26-41, 2003.

[8] L. Haripriya and M. A. Jabbar, "Role of machine learning in intrusion detection system: Review," in *Proc. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 925-929.

[9] M. B. Subba, S. Biswas, and S. Karmakar, "A neural network based system for intrusion detection and attack classification," in *Proc. 2016 Twenty Second National Conference on Communication (NCC)*, 2016, pp. 1-6.

[10] P. S. Tang, X. L. Tang, and Z. Y. Tao, "Research on feature selection algorithm based on mutual information and genetic algorithm," in *Proc. 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 2014.

[11] S. Aksoy, "Feature reduction and selection," Department of Computer Engineering, Bilkent University, 2008.

[12] B. Kavitha, S. Karthikeyan, and B. Chitra, "Efficient intrusion detection with reduced dimension using data mining classification

methods and their performance comparison," in *Proc. International Conference on Business Administration and Information Processing*, 2010, pp. 96-101.
[13] K. S. Desale and R. Ade, "Genetic algorithm based feature selection approach for effective intrusion detection system," in *Proc. 2015 International Conference on Computer Communication and Informatics (ICCCI)*, 2015, pp. 1-6.
[14] Y. Y. Aung and M. M. Min, "An analysis of random forest algorithm based network intrusion detection system," in *Proc. 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2017, pp. 127-132.
[15] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDDCUP99 dataset," in *Proc. IEEE International Conference on Computational Intelligence for Security & Defense Applications*, 2009.
[16] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *Proc. 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 2017, pp. 635-638.
[17] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 5, pp. 649-659, Sept. 2008.
[18] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5197-5204, 2011.
[19] K. Deb, *An Introduction to Genetic Algorithms*, pp. 293-315, 1999.
[20] S. W. Kwok and C. Carter, "Multiple decision trees," *Machine Intelligence & Pattern Recognition*, vol. 4, pp. 327-335, 2013.
[21] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, Aug. 1998.

Zhiqiang Liu received the B.S. and Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China. From December 2012 to January 2014, he visited Illinois State University at Urbana-Champaign (UIUC) and Portland State University (PSU). He is currently focusing on the application of artificial intelligence in network security, simulation experiments, and data analysis.

Yucheng Shi was born in Shanxi Province, China, in 1994. He received the B.S. degree from the Taiyuan University of Technology (TYUT), in 2017. He is currently pursuing the M.S. degree with the School of Software Engineering, Northwestern Polytechnical University (NPU), Xi'an City, Shannxi Province, China. His research interests include network security, software engineering and artificial intelligence.